

IDA PAPER P-2953

DOCUMENTATION OF THE FORCES
MOBILIZATION MODEL (FORCEMOB)
VERSIONS 3.1 AND 3.2

Volume II: Data Preparation Guide

Eleanor L. Schwartz
James S. Thomason, *Project Leader*

January 1996

19960516 034

Prepared for
Office of the Assistant Secretary of Defense
(Economic Security)

Approved for public release; distribution unlimited.



INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract DASW01 94 C 0054 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

IDA PAPER P-2953

DOCUMENTATION OF THE FORCES
MOBILIZATION MODEL (FORCEMOB)
VERSIONS 3.1 AND 3.2

Volume II: Data Preparation Guide

Eleanor L. Schwartz
James S. Thomason, *Project Leader*

January 1996

Approved for public release; distribution unlimited.



INSTITUTE FOR DEFENSE ANALYSES

Contract DASW01 94 C 0054

Task T-AO6-656

PREFACE

Since June of 1988 the Institute for Defense Analyses (IDA) has been assisting the Department of Defense in developing a systematic process to estimate U.S. stockpile requirements for strategic and critical materials. This paper documents the most recent versions of the Forces Mobilization Model (FORCEMOB), one of the quantitative computer models used in this process.

This paper was prepared in partial fulfillment of the task entitled "National Defense Stockpile Analyses." The task was performed for the Office of the Assistant Secretary of Defense (Economic Security).

While the authors are solely responsible for the substance of this paper, they would like to express their particular appreciation to the IDA reviewers, Dr. Lowell Bruce Anderson, Dr. Harry Gilman, and Mr. Stanley Horowitz, as well as to Dr. Paul Halpern of the Office of the Secretary of Defense, for their many constructive suggestions. Thanks are also due to Cori Bradford and Charlene Smith for preparing the manuscript, and to Shelley Smith for fine editorial assistance.

CONTENTS

Preface	iii
I. Overview.....	I-1
A. General Remarks.....	I-1
1. Taxonomy of FORCEMOB Input Files	I-2
2. Structure of This Volume	I-4
B. Data Differences Between Current and Earlier Versions of FORCEMOB	I-4
1. Formatted vs. Binary Files.....	I-5
2. Changes to the Production Process File.....	I-5
3. Changes to the Base Military Requirements File	I-6
4. Changes to the Investment Database Files	I-6
5. Other Changes to the Industry-Level Module Data.....	I-6
6. Miscellaneous Changes.....	I-7
C. The Run File	I-8
1. Functions and Structure	I-8
2. Associating the Run File with the Computer Program.....	I-9
II. The Control Inputs File.....	II-1
A. General Information.....	II-2
1. Run Options	II-2
2. General Format Guides	II-4
3. Some Informative Tables.....	II-5
a. FORCEMOB Input Database Files	II-5
b. Optional FORCEMOB Input Data Files.....	II-7
c. What FORCEMOB Input Files Are Required?	II-8
d. FORCEMOB Output Reports.....	II-10

e. FORCEMOB Subroutines That Read the Control Inputs File	II-14
B. Detailed File Descriptions	II-15
1. Control Inputs File Structure If Both Modules Are To Be Exercised, with Force File Input (Option 0a)	II-16
2. Control Inputs File Structure If Only the Requirements Module Is To Be Exercised, with Force File Input (Option 1a)	II-22
3. Control Inputs File Structure If Only the Industry-Level Module Is To Be Exercised (Option 2)	II-27
4. Control Inputs File Structure If Both Modules Are To Be Exercised, with MEI Requirements File Input (Option 0b).....	II-31
5. Control Inputs File Structure If Only the Requirements Module Is To Be Exercised, with MEI Requirements File Input (Option 1b).....	II-37
C. Sample Control Inputs Files	II-41
1. Sample Control Inputs File for Run Option 0a: Both Modules, Force Structure File	II-42
2. Sample Control Inputs File for Run Option 1a: Requirements Module Only, Force Structure File	II-43
3. Sample Control Inputs File for Run Option 2: Industry-Level Module Only	II-44
4. Sample Control Inputs File for Run Option 0b: Both Modules, MEI Requirements File.....	II-45
5. Sample Control Inputs File for Run Option 1b: Requirements Module Only, MEI Requirements File	II-46
III. Structure and Format of the Database, Optional, and Auxiliary Files.....	III-1
A. Overview	III-1
1. File Selection, Naming, and Location	III-2
2. File Discussion Format	III-3
a. Summary of Data in File Subsection	III-3
b. Record and Format Guide Subsection	III-3
c. Definitions of Symbolic Entities Subsection	III-4
B. The Element Database File	III-5
1. Summary of Data in File.....	III-5
2. Record and Format Guide.....	III-7

3. Definitions of Symbolic Entities	III-8
C. Input Database Files—Requirements Module	III-10
1. Force Structure Database File	III-10
2. Major End Item Inventory File	III-15
3. Cost Database File	III-16
4. Production Process Lead Times File	III-20
5. Production Process Matrix File	III-21
D. Input Database Files—Industry-Level Module	III-23
1. Base Military Requirements Database File	III-23
2. Conflict Military Requirements Database File	III-25
3. Civilian Consumption Requirements Database File	III-27
4. Supply-Side Data on Q/K Ratios and Capacity Utilization Rates	III-29
5. Supply Database File	III-31
6. Investment Distribution File	III-33
7. Investment Lead Times File	III-35
8. Investment Sector Mapping File	III-36
E. Optional FORCEMOB Input Files	III-37
1. Optional File 1—Base Military Factors	III-37
2. Optional File 2—Civilian Factors	III-38
3. Optional File 3—Import/Export Factors	III-39
4. Optional File 4—Major End Item Requirements	III-40
5. Optional File 5—Inventory Allocation	III-42
6. Optional File 6—Military/Civilian Fungibility Factors	III-44
F. FORCEMOB Auxiliary Files	III-45
1. The Debugging Flags File	III-45
2. The Major End Item Aggregation Mapping File	III-47
References	R-1
Glossary	GL-1

FIGURE

I-1	Sample Run File	I-9
-----	-----------------------	-----

TABLES

I-1	FORCEMOB Input Database Files	I-3
I-2	Optional FORCEMOB Input Files	I-4
II-1	FORCEMOB Basic Database Files	II-7
II-2	Optional FORCEMOB Input Data Files.....	II-8
II-3	What FORCEMOB Input Files Are Required?	II-9
II-4	FORCEMOB Output Reports.....	II-12
II-5	What FORCEMOB Routines Read the Control Inputs File?	II-15
III-1	Budget Categories for MEI Aggregation.....	III-47

I. OVERVIEW

A. GENERAL REMARKS

This volume of the documentation of the Forces Mobilization Model (FORCEMOB), Version 3.1, describes the model's input data files. Depending on the options the user selects, up to 25 different data files might be necessary to run FORCEMOB. The main objective of this volume is to present the specific formats of the input data files, so that a programmer or data preparer can construct files that FORCEMOB can read. However, this volume can also give an analyst, user, or model-builder precise information on what kinds of data the model uses. This information can be helpful in determining appropriate data values, understanding the model better, and identifying model elements to be changed, if appropriate.

After the initial version of this documentation was prepared, the FORCEMOB model was modified slightly, and there are now two new versions of it, Versions 3.2a and 3.2b. Version 3.2a was prepared as an interim version of the model for use in the 1995 National Defense Stockpile study; Version 3.2b is the latest version of FORCEMOB. These new versions—including the differences in their input data structure from Version 3.1—are documented in Chapter V of Volume I, which the data preparer should read. The current volume is to be considered as applying to Version 3.1, and the phrase "current version" used in this volume means Version 3.1. However, most of the description, and all of the description for the Industry-level module, also applies to Versions 3.2a and 3.2b. Version 3.2b has an additional optional input file (used in the Requirements module) and an additional output report.

Version 3.1 of FORCEMOB is substantially different from Version 1.0, which was documented in IDA Paper P-2716 [1, 2, 3, 4]. Many of the input data files have been changed extensively from Version 1.0, some have undergone slight changes, and some are unchanged. An interim version of FORCEMOB, Version 2.2, was used in the 1993 National Defense Stockpile study [5], and a more recent interim version, Version 3.0, was used in a study of Navy logistics [6]. Where appropriate, we explain the differences in the data between the current and previous versions.

We caution the reader that most of the input files must follow quite specific formats. Many different formats are used, depending on the particular inputs. The number of data records and their ordering can depend on the values of certain previously read inputs. We assume that the reader of the format descriptions is familiar with FORTRAN programming; FORTRAN notation for variables and formats appears frequently. Other readers should be familiar with FORTRAN's conventions for variable and array names. The use of FORTRAN symbolic names provides a bridge between the precision of the computer code and the more general model description in Volume I of this paper.

Determining appropriate data sources is a separate issue from describing the data files, and this volume does not address it, except insofar as precise knowledge of the inputs affects the form of the data to be used. Data values will most likely vary with the particular study or application for which FORCEMOB is being performed. In previous studies, supply-side data have often been generated from the University of Maryland's INFORUM economic models [7, 8, 9]. Force structure and force requirements data have often been obtained from the Joint Chiefs of Staff, or from the particular client for a study.

1. Taxonomy of FORCEMOB Input Files

The FORCEMOB input files can be divided into five groups:

The Run file (sometimes called the *Master Inputs file*). This file is short but significant. It provides the link between the program and the rest of the data. The Run file contains the names of one or more Control Inputs files.

The Control Inputs file. This file plays a pivotal role in FORCEMOB. Each run of FORCEMOB uses a different Control Inputs file. The file specifies the names of all the database files and optional files to be used in the run. It can also contain factors to modify the values in some of those files, to enable the user to easily run variations off a base case. In addition, the file contains important general information, such as scenario dates.

Basic database files. These files constitute the bulk of the FORCEMOB inputs. Typically, the user prepares a few of each of these files and then runs variations by choosing optional files or by varying the information in the Control Inputs file. Note that although we often use the term "database" files, the information in these files is *not*

handled via a relational database management system package (such as dBase). Table I-1 lists these files.

Table I-1. FORCEMOB Input Database Files

Element File (contains names of weapon types and industry sectors)
Force Structure File
Major End Item Inventory File
Cost Database File
Production Process Lead Times File
Production Process Matrix File
Base Military Requirements File
Conflict Military Requirements File
Civilian Requirements File
Capital/Output Ratios and Capacity Utilization File
Supply Side Data File
Investment Distribution Matrix File
Investment Lead Times File
Investment Sector Mapping File

Optional data files. The user can invoke these files to exercise certain options or data modifications. Table I-2 lists them.

Auxiliary data files. These files are also optional, but are not handled in the same manner as the "optional" data files. They include the Debugging Flags file and the Major End Item Aggregation Mapping file.¹

¹ Recall from Volume I that the "Major End Items" constitute the weapons, consumables, and other military entities necessary to support combat. The set of Major End Items is one of the fundamental inputs to FORCEMOB.

Table I-2. Optional FORCEMOB Input Files

Base Military Factors
Civilian Factors
Import/Export Factors
Major End Item Requirements
Inventory Allocation
Military/Civilian Fungibility Factors

2. Structure of This Volume

The structure of this volume is in concordance with the taxonomy of the input files. Chapter II discusses the Control Inputs file in detail. Chapter III describes the database, optional, and auxiliary data files. The remainder of Chapter I provides:

- A summary of differences between the input data structure of the current FORCEMOB version and that of previous versions
- A description of the Run file

B. DATA DIFFERENCES BETWEEN CURRENT AND EARLIER VERSIONS OF FORCEMOB

As stated earlier, many of the input data files for Version 3.1 of FORCEMOB differ substantially from those for Version 1.0. This section summarizes, in list form, the main differences (the interim versions are mentioned where significant). Additional differences are discussed in the individual sections on each file, in Chapter III. This section is intended for readers who are familiar with previous FORCEMOB versions, and can be skipped if desired.

Some of these differences arise out of methodological changes to FORCEMOB. Some of these differences restrict the generality of the data input, but allow FORCEMOB to run more smoothly and quickly. Furthermore, former data did not take advantage of these now-restricted generalities.

The differences fall into several groups, as indicated by the subsections of this section.

1. Formatted vs. Binary Files

Versions 1 and 2 had parallel series of formatted and binary database files. Each database file existed in both a formatted and a binary form. The user prepared a formatted version, and an auxiliary program, called CONVERTDB2, prepared the corresponding binary version, which FORCEMOB then read. Now, all input database files are formatted. It takes a bit longer for FORCEMOB to read them, but this is balanced by the additional convenience of not maintaining two sets of data files. Also, due to other changes (see below) many of the input files are shorter than previously, so the time savings of a binary read would be less.

In these parallel series, all formatted files had the extension '.FORM' and the binary files had an extension that depended on the type of file. Now, these latter extensions are used for the formatted files (see Table II-1 in Chapter II for the extensions).

The CONVERTDB2 program, in addition to converting formatted to binary files, adjusted certain data for inflation. Now, there is no such adjustment, and the user will have to make sure, when preparing the data files, that the monetary data in all files are in the same year dollars. Related to this, the Inflation Factors file of previous versions is not relevant to Version 3, and the line indicating the name of the Inflation Factors file has been removed from the Element database file.

2. Changes to the Production Process File

The Production Process file was a very large file with an indexed sequential access organization. Reading this file was a major cause of FORCEMOB Version 1.0's long running time. With only slight methodological restrictions, we have been able to replace this big file with two somewhat smaller files, the Production Process Lead Times file and the Production Process Matrix file. The older methodology *allowed* (but the data in fact did not specify) a different requirement amount for each month of the production lead time, for each given Major End Type and industry sector. Now, a total requirement amount is input; the code apportions it evenly over the lead time. (Version 2.2 had the newer methodology, but there was just one data file, in binary format, prepared by a special preprocessor program.)

Version 1.0 allowed two Production Process files to be used, one for base military calculations and one for conflict military calculations. Those base military calculations are no longer performed, as base military requirements on industry are now read in

directly from a file (see section 3, below). The current Production Process Lead Times and Matrix files apply to the calculation of conflict military requirements only.

3. Changes to the Base Military Requirements File

In earlier FORCEMOB versions, there were two choices for organizing the Base Military Requirements file. The requirements could be input for each Major End Item, the FORCEMOB code then computing the requirements on industry. Alternatively, the input file could specify the requirements on industry directly. Now, only the second option is allowed (we might restore the first option in a later program version).

Similarly, the Optional Base Military Factors file previously could have two forms of organization, corresponding to the Base Military Requirements file: factors organized by MEI or factors organized by industry. Now, only the organization by industry is allowed.

The two different choices for organizing the Base Military Requirements file also affected the Major End Item inventories available to offset conflict demands. In the first case, the inventories were computed; in the second, they were input directly. Now, only the second option is allowed (we might restore the first option in a later program version).

The Major End Item inventories, which are relevant for FORCEMOB's Requirements module, now appear in a separate file from the Base Military Requirements, which are used by FORCEMOB's Industry-level module. In Versions 1 and 2 (in the second organization option), these quantities were in the same file.

4. Changes to the Investment Database Files

The methodology behind FORCEMOB's investment modeling has changed considerably from Version 1.0. The current methodology is discussed in Volume I of this paper. These methodological differences are reflected in differences in the input data files. Now, instead of one investment database file, there are three, and the contents are very different from the old file. (Version 2.2 had the same investment methodology as the current version, but the input files had fixed, hard-coded names and needed to be located in the directory from which the program was being run.)

5. Other Changes to the Industry-Level Module Data

The supply-side data, which previously appeared in one file, has been broken up into two files:

- A file of capital/output ratios and plant capacity utilization fractions
- A file with domestic supply, imports, and exports.

A user might frequently wish to run sensitivity cases by varying the supply, import, and/or export data, but might change the capital/output ratios and capacity utilization rates far less often. The two-file structure was developed to facilitate running such sensitivity cases.

In the Base Military Requirements, Civilian Requirements, and Supply databases, data that previously were input on a monthly basis are now input on a yearly basis. The code divides the yearly value by 12 to obtain a monthly value. (Previous code versions *allowed* monthly-varying inputs, but the data in fact did not vary by month within a year.) This change greatly reduces the size of these files.

A new file, Conflict Military Requirements, has been added. It is used as an input file when only the FORCEMOB Industry-level module is being exercised. The data in it, and the file itself, are outputs of FORCEMOB's Requirements module.

An additional optional file, Military/Civilian Fungibility Factors, has been added. This file contains data relevant for the "dual use" methodology that has been developed for FORCEMOB Version 3, as described in Volume I, Chapter II of this paper. (Version 3.0 did not have a formal input file established for these factors; they were read in by interim code.)

6. Miscellaneous Changes

In addition to the preceding categories of changes, the following assorted changes have been made:

- Previously, much of the data in the input files had to be formatted in very specific ways, to be compatible with the FORTRAN READ statements in the computer program. Now, many of the READ statements have been changed to list-directed (*) format, allowing somewhat more flexibility in the arrangement of the data values on the lines of the input files. Careful attention to format is still necessary, however, and this volume provides a record and format guide for each data file.
- The list-directed format requires that data values be separated by spaces or commas. The fixed-column format does not have this requirement. The user should recheck all fixed-column format files and insert spaces (or commas) where necessary before using these files with the new program version.

- An auxiliary postprocessor program² to FORCEMOB Version 1 created a table of Major End Item demands aggregated by Service and DoD budget categories. Now, these computations are performed in FORCEMOB itself—and the Major End Item Aggregation Mapping file that the postprocessor used is now an (auxiliary) input file for FORCEMOB. It is described in Chapter III, Section F.2.
- The Debugging Flags file has the same format as in Version 1.0, but most of its variables are now not used.
- In Versions 1 and 2, one large FORCEMOB subroutine, Subroutine LOAD, read all the database files and optional files. Version 3.1 (and 3.0) has broken LOAD into several shorter, more manageable subroutines, each of which reads a few (or one) of the input files (see Table II-1 in Chapter II).

C. THE RUN FILE

1. Functions and Structure

Although short, the Run file provides the actual link between the data and the FORCEMOB computer program. One execution of the computer program consists of one or more “runs” of the FORCEMOB model. Each run is associated with a Control Inputs file, which in turn specifies the other data files for the run. The Run file simply lists, one per line, the names of the Control Inputs files for the runs. The file can be prepared with a text editor. During file preparation, the following points should be observed about each Control Inputs file name:

- It should *not* be delimited in quotes (the FORCEMOB computer program reads the name in an (A) format).
- It should be a legal DOS name—eight or fewer characters.
- Any extension can be used (we have usually used ‘.IN’).
- A full path specification can be used if desired; otherwise the program looks for the file in the directory from which the program is being run.
- The total length, including the path, should not exceed 80 characters.
- It should be left-justified.

² Program AGGTAB, described in Chapter X of the FORCEMOB Version 1.0 *Users Guide* [2].

In keeping with DOS conventions, the name of the Run file itself should be eight or fewer characters long. No specific extension is required, but in IDA's work on FORCEMOB, we have frequently used the extension '.INF'.

A sample Run file appears in Figure I-1.

TEST1.IN C:\GOODRUNS\RUN1A.IN BIGFORCE.IN

Figure I-1. Sample Run File

2. Associating the Run File with the Computer Program

In order to execute FORCEMOB, the program must ascertain the name of the Run file, and then read the file. The current PC version of FORCEMOB has been compiled with the Microsoft FORTRAN PowerStation compiler [10], which allows command line arguments. In the PC version of FORCEMOB, there is one command line argument—the name of the Run file. Thus if the program executable code file is named FORCEMOB.EXE, then one would enter the command line

FORCEMOB SAMPLE.INF

to run FORCEMOB with the Run file SAMPLE.INF. A full path name for the Run file can be used, as long as the total length does not exceed 80 characters.

In the current VAX version, the name of the Run file is read from standard input—the keyboard or a command procedure file—at the start of the program (as in the PC version, a full path name can be used). In general, converting FORCEMOB to other machines or compilers will involve changing the way the program reads the Run file. The FORCEMOB code has been structured so that the necessary changes will most likely be in the main program (located in file MAIN.FOR).

II. THE CONTROL INPUTS FILE

The objective of this chapter is to describe the FORCEMOB Control Inputs file in sufficient detail that the user can prepare one. The Control Inputs file is a short ASCII file that can be prepared with any text editor. Each entry in the file is discussed, with information on both its format and its meaning. Usually, we list and define the associated computer program variables.

As stated earlier, this file plays a pivotal role in FORCEMOB: each run of the FORCEMOB model is associated with exactly one Control Inputs file. The file contains:

- start and end dates for the scenario period
- names of the input database and optional files to be used
- conflict dates for each theater, as applicable
- factors and multipliers that can be used to change the data in the input files, to facilitate sensitivity analyses
- assorted other parameters used by the model
- a list of output reports desired

At the end of the file, the user can put in comments as appropriate.

In Version 1.0 of FORCEMOB, the information that now appears in the Control Inputs file was typed into forms on the screen (as described in the FORCEMOB Version 1.0 *Users Guide* [2]). These forms were developed in and run as a part of the DEC VAX Forms Management System. For this reason, Version 1.0 is interactive and can be run only on a VAX. Putting this information in an ASCII file facilitates transfer of the program to other types of computers and allows FORCEMOB to be run in batch mode.

IDA has recently developed a user interface for FORCEMOB Version 3.1. It runs on a PC under Windows. The user interface, an interactive program,

- displays screens that request the information of the Control Inputs file; the user types in this information.
- generates the corresponding Control Inputs file.
- runs FORCEMOB.

- allows on-screen examination of FORCEMOB output files.
- can run the Stockpile Sizing module [11] on the stockpile postprocessor file that FORCEMOB generates.

The user interface is documented under separate cover. This chapter focuses on the construction of FORCEMOB's Control Inputs file.

The rest of this chapter is structured as follows. The heart of the chapter is Section B, which presents line by line descriptions of the Control Inputs file. But first, Section A gives some information and tables that aid in interpreting Section B. Section C shows some sample Control Inputs files. While reading Section B, the user might wish to look at these examples.

In keeping with DOS conventions, the Control Inputs file should have a name of eight or fewer characters. No specific extension is required, but in IDA's work on FORCEMOB, we have frequently used the extension '.IN'.

A. GENERAL INFORMATION

1. Run Options

In discussing the Control Inputs file, it is helpful to distinguish five different "run options." The format of the file is somewhat different for each option. The run options arise from FORCEMOB's module structure. As discussed in Volume I of this paper, FORCEMOB consists of two distinct modules, the Requirements module and the Industry-level module. The Requirements module determines the weapons required for a conflict and translates the weapon demand to a demand on industry. The Industry-level module:

1. takes the conflict military demand on industry generated by the Requirements module.
2. adds to it civilian demand and base military demand.
3. compares total demand against supply and determines the supply shortfalls, if any.
4. models the process of investment to redress shortfalls, if there are any.

A given run of FORCEMOB can exercise either or both of the modules, and one of the first entries on the Control Inputs file is a "code section indicator" that specifies which modules to run, as follows:

0—run both modules

1—run the Requirements module only (the conflict military demands on industry are written out to a file, which can then be read by the Industry-level module in another FORCEMOB run)

2—run the Industry-level module only

Within the Requirements module, there are two distinct ways to establish the Major End Item requirements. The “standard” way is to compute the demand from a conflict scenario:

1. Several different kinds of force units arrive at times specified by the user (via the Force Structure database file and Control Inputs file)
2. Each of these units generates a set of demands for various types of weapons and consumables (based on data in the Force Structure database file)
3. The total weapon requirements are computed, multiplied by prices (from the Cost database file), and aggregated into (time phased) dollar demand for Major End Items.

Alternatively, the dollar demand for MEIs can be read directly from an input file. This file is identified as Optional File 4. If the Control Inputs file specifies that Optional File 4 is to be used, then certain information on conflict scenario specification is unnecessary—and the subsequent structure of the Control Inputs file is affected.

Based on the module structure and the method of specifying the MEI requirements, we can thus distinguish five different run options, as follows:

- **Option 0a.** Exercise both the Requirements module and the Industry-level module; MEI requirements are determined from a Force Structure file.
- **Option 1a.** Exercise the Requirements module only; MEI requirements are determined from a Force Structure file.
- **Option 2.** Exercise the Industry-level module only.
- **Option 0b.** Exercise both the Requirements module and the Industry-level module; MEI requirements are read in from an MEI Requirements file (Optional File 4).
- **Option 1b.** Exercise the Requirements module only; MEI requirements are read in from an MEI Requirements file.

As stated earlier, the format of the Control Inputs file is somewhat different for each of the run options. Section B contains a separate format description for each option.

2. General Format Guides

Following are some points to note about how certain information is coded within the Control Inputs file:

- a. All values on the Control Inputs file are read in FORTRAN list-directed (*) format. Values should be separated by blank spaces (or commas).
- b. In particular, all character strings, including COMMENTS listed below, are read in FORTRAN list-directed (*) format, and thus must be enclosed in single quotes.
- c. Additional comments can go at ends of lines, beyond where the program will attempt to read. Such additional comments need not be enclosed in quotes.
- d. All year values have four digits (e.g., 1993). All month values are encoded as 1=January, 2=February, and so forth.
- e. For Yes or No inputs, use the single-quote-delimited single characters 'Y' or 'N'. The program will also accept lowercase 'y' and 'n'.
- f. Except as noted, percentages are integer values, and can range from 0 to 100.
- g. When an input data file is listed, the file *name*, with no extension, should appear. File names should be eight or fewer characters long. The program supplies a specific extension that depends on the type of file. Table II-1, which appears in section A.3, below, shows these extensions. Input files (except for the Control Inputs file and the Debugging Flags file, if present), are assumed to reside in the data file directory specified in the Control Inputs file (see the structure descriptions).
- h. Similarly, when an output report is requested, the file *name* (eight characters or fewer) should be specified, without an extension. The program supplies a specific extension that depends on the type of report. Table II-4, which appears in section A.3, below, shows these extensions. Output files will be put in the output file directory specified in the Control Inputs file (except for the Conflict Military Requirements file, which is put in the data file directory).
- i. When a data file is to be read, the data file directory specification, as it appears in the Control Inputs file, is concatenated with the data file name and the reserved extension. The resultant character string is used as the name of a file, and the FORCEMOB computer program attempts to open that file. This character string must thus be in the correct syntax for file specification, for the particular computer platform being used. To ensure this, the data file directory specification in the Control Inputs file must end with a backslash, when running FORCEMOB on the PC. On the VAX (under VMS), it should end with a closing bracket ("]"), or should consist of a VAX logical name

followed by a colon. Similar considerations apply to the output file directory.

3. Some Informative Tables

This section presents several tables of information on FORCEMOB files. The Control Inputs file descriptions in section B refer to some of these tables—and some of the tables will probably make more sense after section B is read. Each table, along with some explanation and commentary, appears in a separate subsection. The contents of the tables are as follows:

- Table II-1: Selected information about the main database files
- Table II-2: Selected information about the optional data files
- Table II-3: Which database files are required under the various run options?
- Table II-4: Selected information about the FORCEMOB output reports
- Table II-5: FORCEMOB subroutines that read the Control Inputs file

a. FORCEMOB Input Database Files

Table II-1 presents some information about the main database files. The columns—and contents—of Table II-1 are as follows:

1. *File number.* The Control Inputs file refers to these file numbers to identify the type of database file being considered. Admittedly, the numbering is somewhat arbitrary, but it is important that the correct number appear in the Control Inputs file.
2. *Description.* Here, this is just the name used to refer to the type of file. For detailed descriptions of the files and their formats, see Chapter III.
3. *Module.* Most files are associated with either the Requirements module or the Industry-level module, but not both. This column identifies the module (see the discussion below for exceptions).
4. *Extension.* A database file of the given type is always assumed to have the extension given here. FORCEMOB will look for a file with this extension.
5. *Default file name.* If a particular file is needed for a FORCEMOB run (see Table II-3) and no file is specified on the Control Inputs file, then FORCEMOB looks for a file with the default file name specified here and the extension given in the previous column (in the data file directory indicated in the Control Inputs file).

6. *Variable name.* The name of each input data file is a character variable in the FORCEMOB computer program—the name of this variable is shown here.
7. *Subroutine where read.* This is the subroutine of the FORCEMOB computer program that reads the file. This information might be helpful for debugging, if necessary.

The reader should bear in mind the following points:

- The name of the Element database file is hard-coded as ELEMENT.DB. The FORCEMOB computer program looks for a file with this name in the data file directory specified on the Control Inputs file. In order to have several different possible Element files available for use, these files must be located in different directories. It is anticipated that the Element file will change infrequently. Moreover, changes in the Element file will necessitate changes in a number of other data files, and these changed files should be relocated to a different directory to avoid confusion. The use of a hard-coded name for the Element file is a way of enforcing this.
- Since the Element file is always required and always has the same name, it is not listed in the Control Inputs file (it doesn't need to be) and does not require a file number or variable name.
- The Conflict Military Requirements file is an output of the Requirements module and an input to the Industry-level module.
- In the current version of FORCEMOB, the Base Military Requirements file is an input to the Industry-level module. In previous FORCEMOB versions, however, the file had two alternative formats, one of which was connected with the Requirements module. See the discussion in Chapter I, section A.3 and Chapter III, section D.1.

Table II-1. FORCEMOB Basic Database Files

File No.	Description	Module	Extension	Default File Name	Variable Name	Subroutine Where Read
—	Element Database	All	.DB	ELEMENT	—	RDELEM
1	Force Structure	Requirements	.FRC	FORCE	FSFILE	RDFORCE
2	MEI Inventories	Requirements	.MIN	MEIINVN	MINFILE	RDRQMF
3	Cost Data	Requirements	.CST	COSTDAT	CSTFILE	RDRQMF
4	Production Process Lead Times	Requirements	.PPL	PROCLDT	PPLFILE	RDPDSF
5	Production Process Matrix	Requirements	.PPM	PROCMTX	PPMFILE	RDPDSF
6	Base Military Requirements	Industry-level	.MIL	BASEMIL	BMLFILE	RDILMF
7	Conflict Military Requirements	Special (see text)	.CFM	CONFMIL	CFMFILE	RDCFM
8	Civilian Requirements	Industry-level	.CIV	CIVREQ	CIVFILE	RDILMF
9	Q/K Ratios and EOC Fractions	Industry-level	.QKF	QKRLFF	QKFFILE	RDILMF
10	Supply Side Data	Industry-level	.SUP	SUPPLY	SUPFILE	RDILMF
11	Investment Distribution	Industry-level	.IDR	CMAT	IDRFILE	INVSHRT
12	Investment Lead Times	Industry-level	.ILT	GREEN	ILTFILE	INVSHRT
13	Investment Sector Mapping	Industry-level	.ISM	CAPIND	ISMFILE	INVSHRT

b. Optional FORCEMOB Input Data Files

Table II-2 presents some information about the optional data files. The information is like that of Table II-1.

Table II-2. Optional FORCEMOB Input Data Files

Optional File No.	Description	Module	Extension	Variable Name	Subroutine Where Read
1	Base Military Factors	Industry-level	.FBM	OPTFILE(1)	RDFACT
2	Civilian Factors	Industry-level	.FCV	OPTFILE(2)	RDFACT
3	Import/Export Factors	Industry-level	.FIE	OPTFILE(3)	RDFACT
4	MEI Requirements	Requirements	.MEI	OPTFILE(4)	RDMEIRQ
5	Inventory Allocation	Requirements	.INA	OPTFILE(5)	RDINA
6	Military/Civilian Fungibility Factors	Industry-level	.MCF	OPTFILE(6)	RDMCF

c. What FORCEMOB Input Files Are Required?

A run of the FORCEMOB model requires many input data files—but the particular types of files required depend on the run option being exercised. For each run option, Table II-3 shows the types of database files required and the optional files that can be used.

Keep in mind the following points:

1. If a particular type of file is needed for a FORCEMOB run and no such file is specified on the Control Inputs file, then FORCEMOB looks for a file with the default file name and extension specified in Table II-1 (in the data file directory indicated in the Control Inputs file).
2. If the user inadvertently specifies a data file that is of a type not needed for the particular run option, then FORCEMOB does not attempt to find or read the file. The history file for the run will show the data files actually read and used.
3. If the user does not want to exercise the investment portion of the model (this option is specified in the Control Inputs file), or if investment is not needed because there is no shortfall, then the three investment-related database files are not read and are not necessary.

Table II-3. What FORCEMOB Input Files Are Required?

File No.	Description	Run Option				
		0a	0b	1a	1b	2
—	Element Database	X	X	X	X	X
1	Force Structure	X		X		
2	MEI Inventories	X	X	X	X	
3	Cost Data	X	X	X	X	
4	Production Process Lead Times	X	X	X	X	
5	Production Process Matrix	X	X	X	X	
6	Base Military Requirements	X	X			X
7	Conflict Military Requirements					X
8	Civilian Requirements	X	X			X
9	Q/K Ratios and EOC Fractions	X	X			X
10	Supply Side Data	X	X			X
11	Investment Distribution	X	X			X
12	Investment Lead Times	X	X			X
13	Investment Sector Mapping	X	X			X
Optional 1	Base Military Factors	O	O			O
Optional 2	Civilian Factors	O	O			O
Optional 3	Import/Export Factors	O	O			O
Optional 4	MEI Requirements		X		X	
Optional 5	Inventory Allocation	O	O	O	O	
Optional 6	Military/Civilian Fungibility Factors	O	O			O

X — required file O — optional file

d. FORCEMOB Output Reports

FORCEMOB can generate a number of different output reports; the reports present various information and results computed by the model. The particular selection of reports generated is determined by the user, via a section of the Control Inputs file. Table II-4 presents some information about how to format these requests (also see section B). The contents of Table II-4 are as follows:

1. *Report type number.* The Control Inputs file refers to these numbers to identify the type of output report being requested. Admittedly, the numbering is somewhat arbitrary, but it is important that the correct number appear in the Control Inputs file.
2. *Short description.* A brief description is shown here.
3. *Module.* Each output report is associated with either the Requirements module or the Industry-level module—not both. This column identifies the module. If the user requests a report that is not associated with a module being exercised in the current run option, an informative message is printed.
4. *Extension.* The output report will be written to a file. The name of this file is requested on the Control Inputs file and the extension is shown here. (The file will be located in the output file directory specified on the Control Inputs file [but see point 3 on the next page].)
5. *Contents of auxiliary line.* Many of the output report requests contain not only the main request line (file specification line; see section B) but also an auxiliary line that specifies additional information necessary to generate the report. This line should appear just after the main request line. The information on the line is read in FORTRAN list-directed (*) format. Years should be coded as four digits (e.g., 1994).

The following points are also relevant.

1. If the file for an output report has the same name, extension, and directory as a file that already exists, the action taken depends on the “file overwrite indicator,” which appears near the top of the Control Inputs file. If this indicator is 1, the pre-existing file is overwritten. Otherwise, an informative message is printed (on the run history file) and the output report is not generated.
2. Most output reports are available in two separate forms, a space-delimited version suitable for printing and a comma-delimited version that can be read into a spreadsheet package such as Microsoft Excel. These versions have different report code numbers, as shown in the table.

3. The Conflict Military Requirements file (report 10) is an output of the Requirements module, but becomes an input data file for the Industry-level module. For that reason, it is written to the data file directory (specified on the Control Inputs file), not the output file directory.
4. Under run options 0a and 0b (which exercise both modules), the user must request that the Conflict Military Requirements file be generated. Under run options 1a and 1b (which exercise only the Requirements module), the file is generated automatically, but the user can put an explicit request line into the Control Inputs file to specify a name for the Conflict Military Requirements file. Otherwise, the first eight characters of the run title are used.
5. Reports 20 and 24 consist of a table of Major End Item demands aggregated by Service and DoD budget categories. To generate these reports, the auxiliary Major End Item Aggregation Mapping file is necessary. The file must be named AGGMAP.DAT and must reside in the *data* file directory specified in the Control Inputs file. This file is discussed further in Chapter III, section F.2.
6. Report 6, the quarterly postprocessor file, is the vehicle for transferring information from FORCEMOB to the Stockpile Sizing module [11]. The report gives demands by quarter, industry, and category (base military, conflict military, civilian, or investment). The Stockpile Sizing module, a separate computer program from FORCEMOB, takes these demands and uses them to compute demand for materials.
7. All output report requests are processed by Subroutine READOUT of the FORCEMOB computer program. Examining the computer code of Subroutine READOUT can clarify the files involved and the formats of the requests and auxiliary lines.

Table II-4. FORCEMOB Output Reports

Report Type No.	Short Description	Module	Extension	Contents of Auxiliary Line
1	Major End Item units report	Requirements	.UNT	none
2	Major End Item dollars report	Requirements	.DOL	none
3	Supply and demand report by month, for a given year	Industry-level	.SDM	year
4	Supply and demand report by year	Industry-level	.SDY	none
5	Stockpile postprocessor file by year	Industry-level	.PPF	none
6	Stockpile postprocessor file by quarter; will be used by Stockpile Sizing module	Industry-level	.PP2	none
7	Ranked shortfall report	Industry-level	.RSF	none
8	Supply expansion report by month, for a given year	Industry-level	.SXM	year
9	Supply expansion report by year	Industry-level	.SXY	none
10	Conflict Military Requirements file	Requirements	.CFM	none
11	Major End Item units report, comma-delimited	Requirements	.UNC	none
12	Major End Item dollars report, comma-delimited	Requirements	.DLC	none
13	Supply and demand report by month, comma-delimited	Industry-level	.SMC	year
14	Supply and demand report by year, comma-delimited	Industry-level	.SYC	none
15	Stockpile postprocessor file by year, comma-delimited	Industry-level	.PFC	none
16	Stockpile postprocessor file by quarter, comma-delimited	Industry-level	.P2C	none
17	Ranked shortfall report, comma-delimited	Industry-level	.RSC	none
18	Supply expansion report by month, comma-delimited	Industry-level	.XMC	year
19	Supply expansion report by year, comma-delimited	Industry-level	.XYC	none
20	MEI demand aggregated by budget category (AGGTAB report)	Requirements	.AGG	none

Continued

Table II-4. FORCEMOB Output Reports (Continued)

Report Type No.	Short Description	Module	Extension	Contents of Auxiliary Line
21	TOE, consumption and threat items used	Requirements	.TCT	none
22	Ranked end users (MEIs and demand components) for given industries	Requirements	.REU	beginning and ending industries
23	Supply and demand report for all months of simulation (comma-delimited)	Industry-level	.SMS	none
24	MEI demand aggregated by budget category, comma-delimited version	Requirements	.AGC	none
25	TOE, consumption and threat items used, comma-delimited version	Requirements	.TTC	none
26	Ranked end users for given industries, comma-delimited version	Requirements	.RUC	beginning and ending industries
27	(not currently used)			
28	Supply expansion report for all months of simulation, for given industries (comma-delimited)	Industry-level	.XMS	beginning and ending industries
29	Military supply expansion report (comma-delimited)	Industry-level	.MLS	none
30	Supply-side summary spreadsheet-ready output report (comma-delimited)	Industry-level	.TXO	none
31	Force unit delivery profiles	Requirements	.UDP	none
32	MEI demands on industry; amount each MEI demands on a given industry	Requirements	.MDI	beginning and ending industries
33	Industry demands for MEIs; amounts of industry demand one MEI induces	Requirements	.IDM	beginning and ending MEIs
34	Industry requirements for force units; industry requirements induced by each force unit	Requirements	.RQF	beginning and ending industries
35	Force units using MEIs	Requirements	.FM	beginning and ending MEIs

Continued

Table II-4. FORCEMOB Output Reports (Concluded)

Report Type No.	Short Description	Module	Extension	Contents of Auxiliary Line
36	TOE composition for MEIs	Requirements	.TCM	beginning and ending MEIs
37	TOE-MEI correspondence mappings	Requirements	.TMM	none
38	Pre-scenario conflict military demand	Requirements	.PSD	none
39-40	(not currently used)			
41	Force unit delivery profiles, comma-delimited	Requirements	.UDC	none
42	MEI demands on industry, comma-delimited	Requirements	.MIC	beginning and ending industries
43	Industry demands for MEIs, comma-delimited	Requirements	.IMC	beginning and ending MEIs
44	Industry requirements for force units, comma-delimited	Requirements	.RQC	beginning and ending industries
45	Force units using MEIs, comma-delimited	Requirements	.FMC	beginning and ending MEIs
46	TOE composition for MEIs, comma-delimited	Requirements	.TCC	beginning and ending MEIs
47	TOE-MEI correspondence mappings, comma-delimited	Requirements	.TMC	none
48	Pre-scenario conflict military demand, comma-delimited	Requirements	.PSC	none
99	Exit the output routine			

e. FORCEMOB Subroutines That Read the Control Inputs File

As will become apparent in section B, the Control Inputs file has several different sections. Table II-5 indicates which subroutines of the FORCEMOB computer program read these sections. Looking at the computer code of these routines can clarify the FORTRAN formats used for reading, which can in turn clarify the format of the Control

Inputs file. This information might also be useful for interpreting premature stop and other diagnostic messages.

Table II-5. What FORCEMOB Routines Read the Control Inputs File?

Information Section of Control Inputs File	FORCEMOB Subroutine That Reads It
Initial information, scenario dates, directories, data file names, special factors	READ1
Theater parameters on attrition, industry demand, startup costs, inventory sharing (run options 0a and 1a only)	RDTHRP
Theater specification parameters (all Requirements module run options)	RDBAT
MEI Requirements file form and distribution (run options 0b and 1b only)	RMEIFRAC
Supply-side and investment parameters (all Industry-level module run options)	READINV
Desired output files	READOUT

B. DETAILED FILE DESCRIPTIONS

This section presents line by line descriptions of the Control Inputs file, in a tabular format. Since the format of the Control Inputs file is somewhat different for each run option, a separate description, in a separate subsection, appears for each run option. While reading these descriptions, it might be helpful to look at the sample Control Inputs files in section C. Also, the descriptions contain some references to the tables in section A.

Recall that we use the terms "simulation period" and "scenario period" synonymously in this documentation. The "start and end dates of the simulation," as referred to in the following subsections, mean precisely the same thing as the "start and end dates of the scenario period" referred to in Volume I.

1. Control Inputs File Structure If Both Modules Are To Be Exercised, with Force File Input (Option 0a)

Line 1	COMMENT Informative header comment.
Line 2	COMMENT, TITLE Simulation title or identifier label
Line 3	COMMENT, ICODSC Code section indicator (integer). The value here should equal 0, indicating that the full model, i.e., both modules, are to be exercised. Other options are 1 - Exercise Requirements module only. 2 - Exercise Industry-level module only. The value of ICODSC greatly affects the structure of the rest of the Control Inputs file. The structure described here is for the case ICODSC = 0.
Line 4	COMMENT, IFLOWV File overwrite indicator. 0 - Do not overwrite existing output and history files with the same name as a newly requested file. If IFLOWV = 0 and the history file of a run would have the same name as an existing history file, the run terminates. If an output file is requested, an informative message is printed and the file is not generated. 1 - Do overwrite existing output and history files if necessary. If overwrite occurs, an informative message is printed.
Line 5	COMMENT, DDIR Name of subdirectory containing the major input data files. On VAX, a logical name can be used. On PC, DDIR should end with a backslash. Should not exceed 64 characters.
Line 6	COMMENT, OUTDIR Name of subdirectory to which the output report files should be written. On VAX, a logical name can be used. On PC, OUTDIR should end with a backslash. Should not exceed 64 characters. Note: The Conflict Military Requirements file, if requested, will be written to directory DDIR, not OUTDIR.
Line 7	COMMENT, IBEG(1), IBEG(2) Month (1 to 12) and year (4 digits) of start date of simulation.
Line 8	COMMENT, IEND(1), IEND(2) Month and year of end date of simulation.
Line 9	COMMENT, ICBEG(1), ICBEG(2) Month and year of date of start of overall conflict period.

Continued

Control Inputs File Structure—Option 0a (Continued)

Lines 10a, 10b, ...	<p>COMMENT, i, DBFILE(i)</p> <p>Numeral "i" and name of type-i database file to be used (no extension). The program looks in directory DDIR for a file with the specified file name and an extension appropriate for type-i files (see Table II-1). A line should be included for each input file desired. If the program needs a type-i file and no line for a type-i file is input, a default name is assigned to DBFILE(i) and the program looks in directory DDIR for that file. If the program needs a type-i file and cannot find the specified file in directory DDIR, a message is printed and the program stops. If the program does not need a type-i file, any line specifying such a file is ignored. See Table II-3 for the types of files, their numbers, and which files are required. The lines can go in any order, and there can be any number of them. If two or more lines have the same value of i, the name DBFILE(i) specified on the last such line in the Control Inputs file is used.</p>
Line 11	<p>COMMENT, 99, 'XXXXXX'</p> <p>Marker line for end of input file section.</p> <p>Beginning comment, numeral 99 and any dummy character string (in single quotes). The program recognizes the number 99.</p>
Line 12	<p>COMMENT, NOPTFIL</p> <p>Number of optional data files to be used (0 if none). Value of NOPTFIL must equal the number of data lines below this one that are used to specify the optional files.</p>
Lines 13a, 13b, ...	<p>COMMENT, ITYPE, OPTFILE(ITYPE)</p> <p>Optional file type number and name of file (no extension). File type numbers are as follows:</p> <ul style="list-style-type: none"> 1 = Base Military Factors 2 = Civilian Factors 3 = Import/Export Factors 4 = Major End Item Requirements 5 = Inventory Allocation 6 = Military/Civilian Fungibility Factors. <p>The program will look in directory DDIR for the file with the specified name and the appropriate extension (see Table II-2). Lines can be in any order, but there must be exactly NOPTFIL of them.</p> <p>Note: Use of a Major End Item Requirements file (optional file 4) will affect the structure of the rest of the Control Inputs file. This discussion assumes that an MEI Requirements file is NOT being used. See separate description (Option 0b) for Control Inputs file structure if an MEI Requirements file is being used.</p>
Line 14	<p>COMMENT, ICIVFACT</p> <p>ICIVFACT = 1 – set civilian factors to single input value for all industries for a given span of years;</p> <p>ICIVFACT = 0 (or any other value) – do not set the civilian factors thus.</p>

Continued

Control Inputs File Structure—Option 0a (Continued)

Line 14a	<p>CIV, ISCIV, IECIV</p> <p>This line should appear only if value of ICIVFACT is 1 (in line 14).</p> <p>CIV = civilian factor value (real),</p> <p>ISCIV = starting year,</p> <p>IECIV = ending year.</p>
Line 15	<p>COMMENT, IBASFACT</p> <p>IBASFACT = 1 – set base military factors to single input value for all industries for a given span of years;</p> <p>IBASFACT = 0 (or any other value) – do not set the base military factors thus.</p>
Line 15a	<p>BAS, ISBAS, IEBAS</p> <p>This line should appear only if value of IBASFACT is 1 (in line 15).</p> <p>BAS = base military factor value (real),</p> <p>ISBAS = starting year,</p> <p>IEBAS = ending year.</p>
Line 16	<p>COMMENT, LEADF, LDTMIN, LDTMAX</p> <p>These values can be used to alter the Major End Type production lead times from the base values read in from the Production Process Lead Times file. All three values are integer.</p> <p>LEADF = percentage by which to multiply the base lead times (can exceed 100%).</p> <p>LDTMIN = minimum lead time value (months).</p> <p>LDTMAX = maximum lead time value (months).</p> <p>The base lead time is multiplied by LEADF percent and rounded to the nearest integer; the resultant value is then adjusted up to LDTMIN or down to LDTMAX as necessary.</p>
Line 17	<p>COMMENT, (ARULE(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = do account for attrition replacement in that theater;</p> <p>'N' = do not.</p>
Line 18	<p>COMMENT, (AFALL(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = allow that theater to make demands on industry;</p> <p>'N' = do not so allow (then conflict demands in that theater can be satisfied only from existing inventory).</p>
Line 19	<p>COMMENT, (ALOSS(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = do account for unit startup costs as well as attrition losses in that theater;</p> <p>'N' = only account for attrition losses.</p>

Continued

Control Inputs File Structure—Option 0a (Continued)

Line 20	<p>COMMENT, NUMGROUPS</p> <p>Number of inventory sharing groups (0, 1, or 2). Lines 20a and 20b appear only if NUMGROUPS equals 1 or 2; lines 20c and 20d appear only if NUMGROUPS equals 2.</p>
Line 20a	<p>COMMENT, (AGRP1(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = that theater is in inventory share group 1; 'N' = it is not.</p>
Line 20b	<p>COMMENT, (IPRVEC(ITHR), ITHR=1, LNTHR)</p> <p>Four integer values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some characters will not be played (see below). Meaning of values is as follows:</p> <p>0 – that theater is not in inventory share group 1; 1 – that theater has first priority in inventory share group 1; 2 – that theater has second priority in inventory share group 1; and so forth.</p>
Line 20c	<p>COMMENT, (AGRP2(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = that theater is in inventory share group 2; 'N' = it is not.</p>
Line 20d	<p>COMMENT, (IPRVEC(ITHR), ITHR=1, LNTHR)</p> <p>Four integer values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below). Meaning of values is as follows:</p> <p>0 – that theater is not in inventory share group 2; 1 – that theater has first priority in inventory share group 2; 2 – that theater has second priority in inventory share group 2; and so forth.</p>
Line 21	<p>COMMENT</p> <p>Comment line, delimited in single quotes, marks start of theater conflict specification data.</p>

Continued

Control Inputs File Structure—Option 0a (Continued)

Lines 22a 22b 22c 22d	<p>COMMENT, ITHR, APLAY, IPROF(1,ITHR), IPROF(2, ITHR), NMON(ITHR), NMBU(ITHR), PERCENT</p> <p>Four data lines, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) lines must appear, even if some theaters will not be played (see below). For non-played theaters, values must appear, but are ignored.</p> <p>COMMENT (Theater name can go here.)</p> <p>ITHR Number of theater; line 22a must have numeral 1, line 22b, 2, and so forth.</p> <p>APLAY 'Y' = play this theater; 'N' = do not.</p> <p>IPROF(1,ITHR) Month (1-12) of start date of conflict in this theater.</p> <p>IPROF(2,ITHR) Year (four digits) of start date of conflict in this theater.</p> <p>NMON(ITHR) Number of months of conflict in this theater.</p> <p>NMBU(ITHR) Number of months of buildup in this theater.</p> <p>PERCENT Percentage of global inventory allocated to this theater (real value).</p>
Line 23	<p>COMMENT</p> <p>Comment line, delimited in single quotes; marks start of investment data section.</p>
Line 24	<p>COMMENT, FACEOC</p> <p>FACEOC = percentage of gap between peacetime capacity and Emergency Operating Capacity that is to be filled by supply expansion. Same value is used for all industries. Integer value.</p>
Line 25	<p>COMMENT, IRAMP</p> <p>IRAMP = Number of months it takes for a plant to expand from its current operating level to the new level (current level plus FACEOC percent of the spare capacity).</p>
Line 26	<p>COMMENT, ITIME</p> <p>ITIME = 0 – perform the investment algorithm.</p> <p>ITIME = 1 – do not perform the investment algorithm; make no investment.</p> <p>Lines 26a through 26d appear only if the value of ITIME on line 26 is zero.</p>
Line 26a	<p>COMMENT, PERLED</p> <p>PERLED = Percentage value; for each industry, investment times used are the percentage PERLED of the corresponding greenfield investment time. (Can exceed 100%.)</p>
Line 26b	<p>COMMENT, ICOLD</p> <p>ICOLD = Local variable giving the number of months after the simulation start that shortfalls are redressable via investment. The code sets the variable BORDER to ISTART + ICOLD (where ISTART is the starting period of the simulation), and no investment can be completed before period BORDER.</p>
Line 26c	<p>COMMENT, ICONV</p> <p>ICONV = Percentage of shortfall to attempt to meet through investment (integer value). A value of 100 percent is recommended.</p>

Continued

Control Inputs File Structure—Option 0a (Concluded)

Line 26d	<p>COMMENT, MAXITER</p> <p>MAXITER = Maximum number of iterations of investment algorithm; program will end the investment routine if the algorithm has not converged after MAXITER iterations.</p>
Output Report Requests Section	<p>Sequence of "output report requests." Each request consists of one or two lines: a file specification line and, possibly, depending on the report requested, an auxiliary line. There can be an arbitrary number of requests, which can appear in any order.</p> <p>The file specification line of a request has the format COMMENT, IOPT, FNAME</p> <p>The line starts with a comment section (delimited by single quotes) and then contains values for IOPT and FNAME, as defined below:</p> <p>IOPT = (integer) number giving type of output report; see Table II-4.</p> <p>FNAME = file name (eight characters or less, no extension) for output report. Program will provide appropriate extension for the report's type. (If such a file already exists, action proceeds according to the file overwrite parameter, IFLOVW, described above.)</p> <p>The information on the auxiliary line, if any, depends on the report requested. Table II-4 shows this information. List-directed (*) format is used, with no initial comment.</p>
End of Data Record	<p>COMMENT, 99, 'XXXXXX'</p> <p>Optional marker line for end of output report requests section. This line also indicates the end of data in the Control Inputs file. Format is the same as that of the file specification line of an output request, but the numeral 99 is used instead of the output report number. FORCEMOB recognizes 99 as an ending indicator. If the end of data record does not appear, the FORCEMOB run will end when the program encounters the physical end of the Control Inputs file.</p>
Optional Comment Lines	<p>Any comments the user desires can appear after the end of data record. The program will not read these lines. If the end of data record is not used, comment lines should not appear at the end of the file (they will be misread as output report requests).</p>

2. Control Inputs File Structure If Only the Requirements Module Is To Be Exercised, with Force File Input (Option 1a)

Line 1	COMMENT Informative header comment.
Line 2	COMMENT, TITLE Simulation title or identifier label
Line 3	COMMENT, ICODSC Code section indicator (integer). The value here should equal 1, indicating that the Requirements module is to be exercised. Other options are 2 - Exercise Industry-level module 0 - Exercise both modules. The value of ICODSC greatly affects the structure of the rest of the Control Inputs file. The structure described here is for the case ICODSC = 1.
Line 4	COMMENT, IFLOVW File overwrite indicator. 0 - Do not overwrite existing output and history files with the same name as a newly requested file. If IFLOVW = 0 and the history file of a run would have the same name as an existing history file, the run terminates. If an output file is requested, an informative message is printed and the file is not generated. 1 - Do overwrite existing output and history files if necessary. If overwrite occurs, an informative message is printed.
Line 5	COMMENT, DDIR Name of subdirectory containing the major input data files. On VAX, a logical name can be used. On PC, DDIR should end with a backslash. Should not exceed 64 characters.
Line 6	COMMENT, OUTDIR Name of subdirectory to which the output report files should be written. On VAX, a logical name can be used. On PC, OUTDIR should end with a backslash. Should not exceed 64 characters. Note: The Conflict Military Requirements file will be written to directory DDIR, not OUTDIR.
Line 7	COMMENT, IBEG(1), IBEG(2) Month (1 to 12) and year (4 digits) of start date of simulation.
Line 8	COMMENT, IEND(1), IEND(2) Month and year of end date of simulation.
Line 9	COMMENT, ICBEG(1), ICBEG(2) Month and year of date of start of overall conflict period.

Continued

Control Inputs File Structure—Option 1a (Continued)

Lines 10a, 10b, ...	<p>COMMENT, i, DBFILE(i)</p> <p>Numeral "i" and name of type-i database file to be used (no extension). The program looks in directory DDIR for a file with the specified file name and an extension appropriate for type-i files (see Table II-1). A line should be included for each input file desired. If the program needs a type-i file and no line for a type-i file is input, a default name is assigned to DBFILE(i) and the program looks in directory DDIR for that file. If the program needs a type-i file and cannot find the specified file in directory DDIR, a message is printed and the program stops. If the program does not need a type-i file, any line specifying such a file is ignored. See Table II-3 for the types of files, their numbers, and which files are required. The lines can go in any order, and there can be any number of them. If two or more lines have the same value of i, the name DBFILE(i) specified on the last such line in the Control Inputs file is used.</p>
Line 11	<p>COMMENT, 99, 'XXXXXX'</p> <p>Marker line for end of input file section.</p> <p>Beginning comment, numeral 99 and any dummy character string (in single quotes). The program recognizes the number 99.</p>
Line 12	<p>COMMENT, NOPTFIL</p> <p>Number of optional data files to be used (0 if none). Value of NOPTFIL must equal the number of data lines below this one that are used to specify the optional files.</p>
Lines 13a, 13b, ...	<p>COMMENT, ITYPE, OPTFILE(ITYPE)</p> <p>Optional file type number and name of file (no extension). File type numbers are as follows:</p> <ul style="list-style-type: none"> 1 = Base Military Factors 2 = Civilian Factors 3 = Import/Export Factors 4 = Major End Item Requirements 5 = Inventory Allocation 6 = Military/Civilian Fungibility Factors. <p>The program will look in directory DDIR for the file with the specified name and the appropriate extension (see Table II-2). Only optional files 4 and 5 are relevant to the Requirements module; requests for optional files 1, 2, 3, and 6 are ignored. Lines can be in any order, but there must be exactly NOPTFIL of them.</p> <p>Note: Use of a Major End Item Requirements file (optional file 4) will affect the structure of the rest of the Control Inputs file. This discussion assumes that an MEI Requirements file is NOT being used. See separate description (Option 1b) for Control Inputs file structure if an MEI Requirements file is being used.</p>

Continued

Control Inputs File Structure—Option 1a (Continued)

Line 14	<p>COMMENT, LEADF, LDTMIN, LDTMAX</p> <p>These values can be used to alter the Major End Type production lead times from the base values read in from the Production Process Lead Times file. All three values are integer.</p> <p>LEADF = percentage by which to multiply the base lead times (can exceed 100%).</p> <p>LDTMIN = minimum lead time value (months).</p> <p>LDTMAX = maximum lead time value (months).</p> <p>The base lead time is multiplied by LEADF percent and rounded to the nearest integer; the resultant value is then adjusted up to LDTMIN or down to LDTMAX as necessary.</p>
Line 15	<p>COMMENT, (ARULE(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = do account for attrition replacement in that theater;</p> <p>'N' = do not.</p>
Line 16	<p>COMMENT, (AFALL(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = allow that theater to make demands on industry;</p> <p>'N' = do not so allow (then conflict demands in that theater can be satisfied only from existing inventory).</p>
Line 17	<p>COMMENT, (ALOSS(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = do account for unit startup costs as well as attrition losses in that theater;</p> <p>'N' = only account for attrition losses.</p>
Line 18	<p>COMMENT, NUMGROUPS</p> <p>Number of inventory sharing groups (0, 1, or 2). Lines 18a and 18b appear only if NUMGROUPS equals 1 or 2; lines 18c and 18d appear only if NUMGROUPS equals 2.</p>
Line 18a	<p>COMMENT, (AGRP1(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = that theater is in inventory share group 1;</p> <p>'N' = it is not.</p>

Continued

Control Inputs File Structure—Option 1a (Continued)

Line 18b	<p>COMMENT, (IPRVEC(ITHR), ITHR=1, LNTHR)</p> <p>Four integer values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some characters will not be played (see below). Meaning of values is as follows:</p> <p>0 – that theater is not in inventory share group 1; 1 – that theater has first priority in inventory share group 1; 2 – that theater has second priority in inventory share group 1; and so forth.</p>
Line 18c	<p>COMMENT, (AGRP2(ITHR), ITHR=1, LNTHR)</p> <p>Four 'Y' or 'N' (Yes or No) values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below).</p> <p>'Y' = that theater is in inventory share group 2; 'N' = it is not.</p>
Line 18d	<p>COMMENT, (IPRVEC(ITHR), ITHR=1, LNTHR)</p> <p>Four integer values, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) values must appear, even if some theaters will not be played (see below). Meaning of values is as follows:</p> <p>0 – that theater is not in inventory share group 2; 1 – that theater has first priority in inventory share group 2; 2 – that theater has second priority in inventory share group 2; and so forth.</p>
Line 19	<p>COMMENT</p> <p>Comment line, delimited in single quotes, marks start of theater conflict specification data.</p>
Lines 20a 20b 20c 20d	<p>COMMENT, ITHR, APLAY, IPROF(1,ITHR), IPROF(2, ITHR), NMON(ITHR), NMBU(ITHR), PERCENT</p> <p>Four data lines, one for each theater in turn. Four (strictly speaking, the parameter LNTHR) lines must appear, even if some theaters will not be played (see below). For non-played theaters, values must appear, but are ignored.</p> <p>COMMENT (Theater name can go here.) ITHR Number of theater; line 20a must have numeral 1, line 20b, 2, and so forth. APLAY 'Y' = play this theater; 'N' = do not. IPROF(1,ITHR) Month (1-12) of start date of conflict in this theater. IPROF(2,ITHR) Year (four digits) of start date of conflict in this theater. NMON(ITHR) Number of months of conflict in this theater. NMBU(ITHR) Number of months of buildup in this theater. PERCENT Percentage of global inventory allocated to this theater (real value).</p>

Continued

Control Inputs File Structure—Option 1a (Concluded)

Output Report Requests Section	<p>Sequence of "output report requests." Each request consists of one or two lines: a file specification line and, possibly, depending on the report requested, an auxiliary line. There can be an arbitrary number of requests, which can appear in any order.</p> <p>The file specification line of a request has the format COMMENT, IOPT, FNAME</p> <p>The line starts with a comment section (delimited by single quotes) and then contains values for IOPT and FNAME, as defined below:</p> <p>IOPT = (integer) number giving type of output report; see Table II-4. If type number is inappropriate for the Requirements module, a message is printed and the request is ignored.</p> <p>FNAME = file name (eight characters or less, no extension) for output report. Program will provide appropriate extension for the report's type. (If such a file already exists, action proceeds according to the file overwrite parameter, IFLOVW, described above.)</p> <p>The information on the auxiliary line, if any, depends on the report requested. Table II-4 shows this information. List-directed (*) format is used, with no initial comment.</p>
End of Data Record	<p>COMMENT, 99, 'XXXXXX'</p> <p>Optional marker line for end of output report requests section. This line also indicates the end of data in the Control Inputs file. Format is the same as that of the file specification line of an output request, but the numeral 99 is used instead of the output report number. FORCEMOB recognizes 99 as an ending indicator. If the end of data record does not appear, the FORCEMOB run will end when the program encounters the physical end of the Control Inputs file.</p>
Optional Comment Lines	<p>Any comments the user desires can appear after the end of data record. The program will not read these lines. If the end of data record is not used, comment lines should not appear at the end of the file (they will be misread as output report requests).</p>

3. Control Inputs File Structure If Only the Industry-Level Module Is To Be Exercised (Option 2)

Line 1	COMMENT Informative header comment.
Line 2	COMMENT, TITLE Simulation title or identifier label
Line 3	COMMENT, ICODSC Code section indicator (integer). The value here should equal 2, indicating that the Industry-level module is to be exercised. Other options are 1 - Exercise Requirements module 0 - Exercise both modules. The value of ICODSC greatly affects the structure of the rest of the Control Inputs file. The structure described here is for the case ICODSC = 2.
Line 4	COMMENT, IFLOVW File overwrite indicator. 0 - Do not overwrite existing output and history files with the same name as a newly requested file. If IFLOVW = 0 and the history file of a run would have the same name as an existing history file, the run terminates. If an output file is requested, an informative message is printed and the file is not generated. 1 - Do overwrite existing output and history files if necessary. If overwrite occurs, an informative message is printed.
Line 5	COMMENT, DDIR Name of subdirectory containing the major input data files. On VAX, a logical name can be used. On PC, DDIR should end with a backslash. Should not exceed 64 characters.
Line 6	COMMENT, OUTDIR Name of subdirectory to which the output files should be written. On VAX, a logical name can be used. On PC, OUTDIR should end with a backslash. Should not exceed 64 characters.
Line 7	COMMENT, IBEG(1), IBEG(2) Month (1 to 12) and year (four digits) of start date of simulation.
Line 8	COMMENT, IEND(1), IEND(2) Month and year of end date of simulation.

Continued

Control Inputs File Structure—Option 2 (Continued)

Lines 9a, 9b, ...	<p>COMMENT, i, DBFILE(i)</p> <p>Numeral "i" and name of type-i database file to be used (no extension). The program looks in directory DDIR for a file with the specified file name and an extension appropriate for type-i files (see Table II-1). A line should be included for each input file desired. If the program needs a type-i file and no line for a type-i file is input, a default name is assigned to DBFILE(i) and the program looks in directory DDIR for that file. If the program needs a type-i file and cannot find the specified file in directory DDIR, a message is printed and the program stops. If the program does not need a type-i file, any line specifying such a file is ignored. See Table II-3 for the types of files, their numbers, and which files are required. The lines can go in any order, and there can be any number of them. If two or more lines have the same value of i, the name DBFILE(i) specified on the last such line in the Control Inputs file is used.</p>
Line 10	<p>COMMENT, 99, 'XXXXXX'</p> <p>Marker line for end of input file section.</p> <p>Beginning comment, numeral 99 and any dummy character string (in single quotes). The program recognizes the number 99.</p>
Line 11	<p>COMMENT, NOPTFIL</p> <p>Number of optional data files to be used (0 if none). Value of NOPTFIL must equal the number of data lines below this one that are used to specify the optional files.</p>
Lines 12a, 12b, ...	<p>COMMENT, ITYPE, OPTFILE (ITYPE)</p> <p>Optional file type number and name of file (no extension). File type numbers are as follows:</p> <ul style="list-style-type: none"> 1 = Base Military Factors 2 = Civilian Factors 3 = Import/Export Factors 4 = Major End Item Requirements 5 = Inventory Allocation 6 = Military/Civilian Fungibility Factors. <p>The program will look in directory DDIR for the file with the specified name and the appropriate extension (see Table II-2). Only optional files 1, 2, 3, and 6 are relevant to the Industry-level module; requests for optional files 4 and 5 are ignored. Lines can be in any order, but there must be exactly NOPTFIL of them.</p>
Line 13	<p>COMMENT, ICIVFACT</p> <p>ICIVFACT = 1 – set civilian factors to single input value for all industries for a given span of years;</p> <p>ICIVFACT = 0 (or any other value) – do not set the civilian factors thus.</p>
Line 13a	<p>CIV, ISCIV, IECIV</p> <p>This line should appear only if value of ICIVFACT is 1 (in line 13).</p> <p>CIV = civilian factor value (real),</p> <p>ISCIV = starting year,</p> <p>IECIV = ending year.</p>

Continued

Control Inputs File Structure—Option 2 (Continued)

Line 14	<p>COMMENT, IBASFACT</p> <p>IBASFACT = 1 – set base military factors to single input value for all industries for a given span of years;</p> <p>IBASFACT = 0 (or any other value) – do not set the base military factors thus.</p>
Line 14a	<p>BAS, ISBAS, IEBAS</p> <p>This line should appear only if value of IBASFACT is 1 (in line 14).</p> <p>BAS = base military factor value (real),</p> <p>ISBAS = starting year,</p> <p>IEBAS = ending year.</p>
Line 15	<p>COMMENT</p> <p>Comment line, delimited in single quotes; marks start of investment data section.</p>
Line 16	<p>COMMENT, FACEOC</p> <p>FACEOC = percentage of gap between peacetime capacity and Emergency Operating Capacity that is to be filled by supply expansion. Same value is used for all industries. Integer value.</p>
Line 17	<p>COMMENT, IRAMP</p> <p>IRAMP = Number of months it takes for a plant to expand from its current operating level to the new level (current level plus FACEOC percent of the spare capacity)</p>
Line 18	<p>COMMENT, ITIME</p> <p>ITIME = 0 – perform the investment algorithm.</p> <p>ITIME = 1 – do not perform the investment algorithm; make no investment.</p> <p>Lines 18a through 18d appear only if the value of ITIME on line 18 is zero.</p>
Line 18a	<p>COMMENT, PERLED</p> <p>PERLED = Percentage value; for each industry, investment times used are the percentage PERLED of the corresponding greenfield investment time. (Can exceed 100%).</p>
Line 18b	<p>COMMENT, ICOLD</p> <p>ICOLD = Local variable giving the number of months after the simulation start that shortfalls are redressable via investment. The code sets the variable BORDER to ISTART + ICOLD (where ISTART is the starting period of the simulation), and no investment can be completed before period BORDER.</p>
Line 18c	<p>COMMENT, ICONV</p> <p>ICONV = Percentage of shortfall to attempt to meet through investment (integer value). A value of 100 percent is recommended.</p>
Line 18d	<p>COMMENT, MAXITER</p> <p>MAXITER = Maximum number of iterations of investment algorithm; program will end the investment routine if the algorithm has not converged after MAXITER iterations.</p>

Continued

Control Inputs File Structure—Option 2 (Concluded)

Output Report Requests Section	<p>Sequence of "output report requests." Each request consists of one or two lines: a file specification line and, possibly, depending on the report requested, an auxiliary line. There can be an arbitrary number of requests, which can appear in any order. The file specification line of a request has the format</p> <p style="margin-left: 40px;">COMMENT, IOPT, FNAME</p> <p style="margin-left: 40px;">The line starts with a comment section (delimited by single quotes) and then contains values for IOPT and FNAME, as defined below:</p> <p style="margin-left: 40px;">IOPT = (integer) number giving type of output report; see Table II-4. If type number is inappropriate for Industry-level module, a message is printed and the request is ignored.</p> <p style="margin-left: 40px;">FNAME = file name (eight characters or less, no extension) for output report. Program will provide appropriate extension for the report's type. (If such a file already exists, action proceeds according to the file overwrite parameter, IFLOVW, described above.)</p> <p>The information on the auxiliary line, if any, depends on the report requested. Table II-4 shows this information. List-directed (*) format is used, with no initial comment.</p>
End of Data Record	<p>COMMENT, 99, 'XXXXXX'</p> <p>Optional marker line for end of output report requests section. This line also indicates the end of data in the Control Inputs file. Format is the same as that of the file specification line of an output request, but the numeral 99 is used instead of the output report number. FORCEMOB recognizes 99 as an ending indicator. If the end of data record does not appear, the FORCEMOB run will end when the program encounters the physical end of the Control Inputs file.</p>
Optional Comment Lines	<p>Any comments the user desires can appear after the end of data record. The program will not read these lines. If the end of data record is not used, comment lines should not appear at the end of the file (they will be misread as output report requests).</p>

4. Control Inputs File Structure If Both Modules Are To Be Exercised, with MEI Requirements File Input (Option 0b)

Line 1	COMMENT Informative header comment.
Line 2	COMMENT, TITLE Simulation title or identifier label
Line 3	COMMENT, ICODSC Code section indicator (integer). The value here should equal 0, indicating that the full model, i.e., both modules, are to be exercised. Other options are 1 - Exercise Requirements module only 2 - Exercise Industry-level module only. The value of ICODSC greatly affects the structure of the rest of the Control Inputs file. The structure described here is for the case ICODSC = 0.
Line 4	COMMENT, IFLOVW File overwrite indicator. 0 - Do not overwrite existing output and history files with the same name as a newly requested file. If IFLOVW = 0 and the history file of a run have the same name as an existing history file, the run terminates. If output file is requested, an informative message is printed and the file is not generated. 1 - Do overwrite existing output and history files if necessary. If overwrite occurs, an informative message is printed.
Line 5	COMMENT, DDIR Name of subdirectory containing the major input data files. On VAX, a logical name can be used. On PC, DDIR should end with a backslash. Should not exceed 64 characters.
Line 6	COMMENT, OUTDIR Name of subdirectory to which the output report files should be written. On VAX, a logical name can be used. On PC, OUTDIR should end with a backslash. Should not exceed 64 characters. Note: The Conflict Military Requirements file, if requested, will be written to directory DDIR, not OUTDIR.
Line 7	COMMENT, IBEG(1), IBEG(2) Month (1 to 12) and year (4 digits) of start date of simulation.
Line 8	COMMENT, IEND(1), IEND(2) Month and year of end date of simulation.
Line 9	COMMENT, ICBEG(1), ICBEG(2) Month and year of date of start of overall conflict period.

Continued

Control Inputs File Structure—Option 0b (Continued)

Lines 10a, 10b, ...	<p>COMMENT, i, DBFILE(i)</p> <p>Numeral "i" and name of type-i database file to be used (no extension). The program looks in directory DDIR for a file with the specified file name and an extension appropriate for type-i files (see Table II-1). A line should be included for each input file desired. If the program needs a type-i file and no line for a type-i file is input, a default name is assigned to DBFILE(i) and the program looks in directory DDIR for that file. If the program needs a type-i file and cannot find the specified file in directory DDIR, a message is printed and the program stops. If the program does not need a type-i file, any line specifying such a file is ignored. See Table II-3 for the types of files, their numbers, and which files are required. The lines can go in any order, and there can be any number of them. If two or more lines have the same value of i, the name DBFILE(i) specified on the last such line in the Control Inputs file is used.</p>
Line 11	<p>COMMENT, 99, 'XXXXXX'</p> <p>Marker line for end of input file section.</p> <p>Beginning comment, numeral 99 and any dummy character string (in single quotes). The program recognizes the number 99.</p>
Line 12	<p>COMMENT, NOPTFIL</p> <p>Number of optional data files to be used (0 if none). Value of NOPTFIL must equal the number of data lines below this one that are used to specify the optional files.</p>
Lines 13a, 13b, ...	<p>COMMENT, ITYPE, OPTFILE(ITYPE)</p> <p>Optional file type number and name of file (no extension). File type numbers are as follows:</p> <ul style="list-style-type: none"> 1 = Base Military Factors 2 = Civilian Factors 3 = Import/Export Factors 4 = Major End Item Requirements 5 = Inventory Allocation 6 = Military/Civilian Fungibility Factors. <p>The program will look in directory DDIR for the file with the specified name and the appropriate extension (see Table II-2). Lines can be in any order, but there must be exactly NOPTFIL of them.</p> <p>Note: Use of a Major End Item Requirements file (optional file 4) will affect the structure of the rest of the Control Inputs file. This discussion assumes that an MEI Requirements file IS being used. See separate description (Option 0a) for Control Inputs file structure if an MEI Requirements file is not being used. If an MEI Requirements file is specified here, then any Force Structure file that might have been specified on Line 10 is ignored.</p>

Continued

Control Inputs File Structure—Option 0b (Continued)

Line 14	<p>COMMENT, ICIVFACT</p> <p>ICIVFACT = 1 – set civilian factors to single input value for all industries for a given span of years;</p> <p>ICIVFACT = 0 (or any other value) – do not set the civilian factors thus.</p>
Line 14a	<p>CIV, ISCIV, IECIV</p> <p>This line should appear only if value of ICIVFACT is 1 (in line 14).</p> <p>CIV = civilian factor value (real),</p> <p>ISCIV = starting year,</p> <p>IECIV = ending year.</p>
Line 15	<p>COMMENT, IBASFACT</p> <p>IBASFACT = 1 – set base military factors to single input value for all industries for a given span of years;</p> <p>IBASFACT = 0 (or any other value) – do not set the base military factors thus.</p>
Line 15a	<p>BAS, ISBAS, IEBAS</p> <p>This line should appear only if value of IBASFACT is 1 (in line 15).</p> <p>BAS = base military factor value (real),</p> <p>ISBAS = starting year,</p> <p>IEBAS = ending year.</p>
Line 16	<p>COMMENT, LEADF, LDTMIN, LDTMAX</p> <p>These values can be used to alter the Major End Type production lead times from the base values read in from the Production Process Lead Times file. All three values are integer.</p> <p>LEADF = percentage by which to multiply the base lead times (can exceed 100%).</p> <p>LDTMIN = minimum lead time value (months).</p> <p>LDTMAX = maximum lead time value (months).</p> <p>The base lead time is multiplied by LEADF percent and rounded to the nearest integer; the resultant value is then adjusted up to LDTMIN or down to LDTMAX as necessary.</p>
Line 17	<p>COMMENT</p> <p>Comment line, delimited in single quotes, marks start of theater conflict specification data.</p>

Continued

Control Inputs File Structure—Option 0b (Continued)

Line 18	<p>COMMENT, ITHR, APLAY, IPROF(1,ITHR), IPROF(2, ITHR), NMON(ITHR), NMBU(ITHR), PERCENT</p> <p>If an MEI Requirements file is being used, there is one data line here, to represent the one dummy theater being modeled.</p> <p>COMMENT Any informative comment, single quoted.</p> <p>ITHR Must equal the numeral "1".</p> <p>APLAY 'Y' = play this theater; 'N' = do not. When an MEI Requirements file is used, a single quoted character must appear here, but the one dummy theater is played regardless.</p> <p>IPROF(1,1) Month (1-12) of start date of conflict in the one dummy theater.</p> <p>IPROF(2,1) Year (four digits) of start date of conflict in the one dummy theater. Note that this start date can be later than the date ICBEQ specified on line 9.</p> <p>NMON(1) Number of months of conflict.</p> <p>NMBU(1) Number of months of buildup.</p> <p>PERCENT Percentage of global inventory allocated to this theater (real value). Can be used to withhold inventory; inventory not allocated cannot help satisfy MEI requirements.</p>
Line 19	<p>Note: Lines 19 and 19a appear only if an MEI Requirements file is used.</p> <p>COMMENT, JREQFLG</p> <p>JREQFLG = 1—requirements specified for each MEI and month of conflict.</p> <p>JREQFLG = 2—requirements specified by MEI only (distribution over months of conflict is specified on line 19a, below).</p> <p>Value of JREQFLG here should be the same as the value of IREQFLG specified in the MEI Requirements file.</p>
Line 19a	<p>Note: Line 19a appears only if an MEI Requirements file is used, and then only if that file specifies MEI requirements by MEI only, not MEI and month (i.e., JREQFLG = 2).</p> <p>COMMENT, (THRPR(IMON), IMON=1, NMON(1))</p> <p>Percentage amount representing percentage of requirement for an MEI that occurs in month IMON of conflict. Same percentages are applied to each MEI. Month IMON ranges from 1 through the value of NMON(1), the number of months of conflict in (dummy) theater 1, which was entered on line 18. Values of THRPR can be continued on subsequent records as necessary.</p>
Line 20	<p>COMMENT</p> <p>Comment line, delimited in single quotes; marks start of investment data section.</p>
Line 21	<p>COMMENT, FACEOC</p> <p>FACEOC = percentage of gap between peacetime capacity and Emergency Operating Capacity that is to be filled by supply expansion. Same value is used for all industries. Integer value.</p>
Line 22	<p>COMMENT, IRAMP</p> <p>IRAMP = Number of months it takes for a plant to expand from its current operating level to the new level (current level plus FACEOC percent of the spare capacity).</p>

Continued

Control Inputs File Structure—Option 0b (Continued)

Line 23	<p>COMMENT, ITIME</p> <p>ITIME = 0 – perform the investment algorithm.</p> <p>ITIME = 1 – do not perform the investment algorithm; make no investment.</p> <p>Lines 23a through 23d appear only if the value of ITIME on line 23 is zero.</p>
Line 23a	<p>COMMENT, PERLED</p> <p>PERLED = Percentage value; for each industry, investment times used are the percentage PERLED of the corresponding greenfield investment time. (Can exceed 100%).</p>
Line 23b	<p>COMMENT, ICOLD</p> <p>ICOLD = Local variable giving the number of months after the simulation start that shortfalls are redressable via investment. The code sets the variable BORDER to ISTART + ICOLD (where ISTART is the starting period of the simulation), and no investment can be completed before period BORDER.</p>
Line 23c	<p>COMMENT, ICONV</p> <p>ICONV = Percentage of shortfall to attempt to meet through investment (integer value). A value of 100 percent is recommended.</p>
Line 23d	<p>COMMENT, MAXITER</p> <p>MAXITER = Maximum number of iterations of investment algorithm; program will end the investment routine if the algorithm has not converged after MAXITER iterations.</p>
Output Report Requests Section	<p>Sequence of "output report requests." Each request consists of one or two lines: a file specification line and, possibly, depending on the report requested, an auxiliary line. There can be an arbitrary number of requests, which can appear in any order.</p> <p>The file specification line of a request has the format</p> <p>COMMENT, IOPT, FNAME</p> <p>The line starts with a comment section (delimited by single quotes) and then contains values for IOPT and FNAME, as defined below:</p> <p>IOPT = (integer) number giving type of output report; see Table II-4.</p> <p>FNAME = file name (eight characters or less, no extension) for output report. Program will provide appropriate extension for the report's type. (If such a file already exists, action proceeds according to the file overwrite parameter, IFLOVW, described above.)</p> <p>The information on the auxiliary line, if any, depends on the report requested. Table II-4 shows this information. List-directed (*) format is used, with no initial comment.</p>

Continued

Control Inputs File Structure—Option 0b (Concluded)

End of Data Record	COMMENT, 99, 'XXXXXX' Optional marker line for end of output report requests section. This line also indicates the end of data in the Control Inputs file. Format is the same as that of the file specification line of an output request, but the numeral 99 is used instead of the output report number. FORCEMOB recognizes 99 as an ending indicator. If the end of data record does not appear, the FORCEMOB run will end when the program encounters the physical end of the Control Inputs file.
Optional Comment Lines	Any comments the user desires can appear after the end of data record. The program will not read these lines. If the end of data record is not used, comment lines should not appear at the end of the file (they will be misread as output report requests).

5. Control Inputs File Structure If Only the Requirements Module Is To Be Exercised, with MEI Requirements File Input (Option 1b)

Line 1	COMMENT Informative header comment.
Line 2	COMMENT, TITLE Simulation title or identifier label
Line 3	COMMENT, ICODSC Code section indicator (integer). The value here should equal 1, indicating that the Requirements module is to be exercised. Other options are 2 - Exercise Industry-level module 0 - Exercise both modules The value of ICODSC greatly affects the structure of the rest of the Control Inputs file. The structure described here is for the case ICODSC = 1.
Line 4	COMMENT, IFLOWV File overwrite indicator. 0 - Do not overwrite existing output and history files with the same name as a newly requested file. If IFLOWV = 0 and the history file of a run would have the same name as an existing history file, the run terminates. If an output file is requested, an informative message is printed and the file is not generated. 1 - Do overwrite existing output and history files if necessary. If overwrite occurs, an informative message is printed.
Line 5	COMMENT, DDIR Name of subdirectory containing the major input data files. On VAX, a logical name can be used. On PC, DDIR should end with a backslash. Should not exceed 64 characters.
Line 6	COMMENT, OUTDIR Name of subdirectory to which the output report files should be written. On VAX, a logical name can be used. On PC, OUTDIR should end with a backslash. Should not exceed 64 characters. Note: The Conflict Military Requirements file will be written to directory DDIR, not OUTDIR.
Line 7	COMMENT, IBEG(1), IBEG(2) Month (1 to 12) and year (4 digits) of start date of simulation.
Line 8	COMMENT, IEND(1), IEND(2) Month and year of end date of simulation.
Line 9	COMMENT, ICBEG(1), ICBEG(2) Month and year of date of start of overall conflict period.

Continued

Control Inputs File Structure—Option 1b (Continued)

Lines 10a, 10b, ...	<p>COMMENT, i, DBFILE(i)</p> <p>Numeral "i" and name of type-i database file to be used (no extension). The program looks in directory DDIR for a file with the specified file name and an extension appropriate for type-i files (see Table II-1). A line should be included for each input file desired. If the program needs a type-i file and no line for a type-i file is input, a default name is assigned to DBFILE(i) and the program looks in directory DDIR for that file. If the program needs a type-i file and cannot find the specified file in directory DDIR, a message is printed and the program stops. If the program does not need a type-i file, any line specifying such a file is ignored. See Table II-3 for the types of files, their numbers, and which files are required. The lines can go in any order, and there can be any number of them. If two or more lines have the same value of i, the name DBFILE(i) specified on the last such line in the Control Inputs file is used.</p>
Line 11	<p>COMMENT, 99, 'XXXXXX'</p> <p>Marker line for end of input file section.</p> <p>Beginning comment, numeral 99 and any dummy character string (in single quotes). The program recognizes the number 99.</p>
Line 12	<p>COMMENT, NOPTFIL</p> <p>Number of optional data files to be used. Value of NOPTFIL must equal the number of data lines below this one that are used to specify the optional files.</p>
Lines 13a, 13b, ...	<p>COMMENT, ITYPE, OPTFILE(ITYPE)</p> <p>Optional file type number and name of file (no extension). File type numbers are as follows:</p> <ul style="list-style-type: none"> 1 = Base Military Factors 2 = Civilian Factors 3 = Import/Export Factors 4 = Major End Item Requirements 5 = Inventory Allocation 6 = Military/Civilian Fungibility Factors. <p>The program will look in directory DDIR for the file with the specified name and the appropriate extension (see Table II-2). Only optional files 4 and 5 are relevant to the Requirements module; requests for optional files 1, 2, 3, and 6 are ignored. Lines can be in any order, but there must be exactly NOPTFIL of them.</p> <p>Note: Use of a Major End Item Requirements file (optional file 4) will affect the structure of the rest of the Control Inputs file. This discussion assumes that an MEI Requirements file IS being used. See separate description (Option 1a) for Control Inputs file structure if an MEI Requirements file is not being used. If an MEI Requirements file is specified here, then any Force Structure file that might have been specified on Line 10 is ignored.</p>

Continued

Control Inputs File Structure—Option 1b (Continued)

Line 14	<p>COMMENT, LEADF, LDTMIN, LDTMAX</p> <p>These values can be used to alter the Major End Type production lead times from the base values read in from the Production Process Lead Times file. All three values are integer.</p> <p>LEADF = percentage by which to multiply the base lead times (can exceed 100%).</p> <p>LDTMIN = minimum lead time value (months).</p> <p>LDTMAX = maximum lead time value (months).</p> <p>The base lead time is multiplied by LEADF percent and rounded to the nearest integer; the resultant value is then adjusted up to LDTMIN or down to LDTMAX as necessary.</p>
Line 15	<p>COMMENT</p> <p>Comment line, delimited in single quotes, marks start of theater conflict specification data.</p>
Line 16	<p>COMMENT, ITHR, APLAY, IPROF(1,ITHR), IPROF(2, ITHR), NMON(ITHR), NMBU(ITHR), PERCENT</p> <p>If an MEI Requirements file is being used, there is one data line here, to represent the one dummy theater being modeled.</p> <p>COMMENT Any informative comment, single quoted.</p> <p>ITHR Must equal the numeral "1".</p> <p>APLAY 'Y' = play this theater; 'N' = do not. When an MEI Requirements file is used, a single quoted character must appear here, but the one dummy theater is played regardless.</p> <p>IPROF(1,1) Month (1-12) of start date of conflict in the one dummy theater.</p> <p>IPROF(2,1) Year (four digits) of start date of conflict in the one dummy theater. Note that this start date can be later than the date ICBEG specified on line 9.</p> <p>NMON(1) Number of months of conflict.</p> <p>NMBU(1) Number of months of buildup.</p> <p>PERCENT Percentage of global inventory allocated to this theater (real value). Can be used to withhold inventory; inventory not allocated cannot help satisfy MEI requirements.</p>
Line 17	<p>Note: Lines 17 and 17a appear only if an MEI Requirements file is used.</p> <p>COMMENT, JREQFLG</p> <p>Indicator for structure of the MEI Requirements file.</p> <p>JREQFLG = 1—requirements specified for each MEI and month of conflict.</p> <p>JREQFLG = 2—requirements specified by MEI only (distribution over months of conflict is specified on line 17a, below).</p> <p>Value of JREQFLG here should be the same as the value of IREQFLG specified in the MEI Requirements file.</p>

Continued

Control Inputs File Structure—Option 1b (Concluded)

Line 17a	<p>Note: Line 17a appears only if an MEI Requirements file is used, and then only if that file specifies MEI requirements by MEI only, not MEI and month (i.e., JREQFLG = 2).</p> <p>COMMENT, (THRPR(IMON), IMON=1, NMON(1))</p> <p>Percentage amount (real value), representing percentage of requirement for an MEI that occurs in month IMON of conflict. Same percentages are applied to each MEI. Month IMON ranges from 1 through the value of NMON(1), the number of months of conflict in (dummy) theater 1, which was entered on line 16. Values of THRPR can be continued on subsequent records as necessary.</p>
Output Report Requests Section	<p>Sequence of "output report requests." Each request consists of one or two lines: a file specification line and, possibly, depending on the report requested, an auxiliary line. There can be an arbitrary number of requests, which can appear in any order.</p> <p>The file specification line of a request has the format</p> <p>COMMENT, IOPT, FNAME</p> <p>The line starts with a comment section (delimited by single quotes) and then contains values for IOPT and FNAME, as defined below:</p> <p>IOPT = (integer) number giving type of output report; see Table II-4. If type number is inappropriate for the Requirements module, a message is printed and the request is ignored.</p> <p>FNAME = file name (eight characters or less, no extension) for output report. Program will provide appropriate extension for the report's type. (If such a file already exists, action proceeds according to the file overwrite parameter, IFLOVW, described above.)</p> <p>The information on the auxiliary line, if any, depends on the report requested. Table II-4 shows this information. List-directed (*) format is used, with no initial comment.</p>
End of Data Record	<p>COMMENT, 99, 'XXXXXX'</p> <p>Optional marker line for end of output report requests section. This line also indicates the end of data in the Control Inputs file. Format is the same as that of the file specification line of an output request, but the numeral 99 is used instead of the output report number. FORCEMOB recognizes 99 as an ending indicator. If the end of data record does not appear, the FORCEMOB run will end when the program encounters the physical end of the Control Inputs file.</p>
Optional Comment Lines	<p>Any comments the user desires can appear after the end of data record. The program will not read these lines. If the end of data record is not used, comment lines should not appear at the end of the file (they will be misread as output report requests).</p>

C. SAMPLE CONTROL INPUTS FILES

This section contains several sample Control Inputs files, one for each run option. The sample files illustrate the variety of options and commenting that are possible. Note that these files were constructed to be used with IDA's FORCEMOB test databases, which have only dummy data and scenario dates in the 1700s (to avoid any potential problem with misinterpretation of current dates).

1. Sample Control Inputs File for Run Option 0a: Both Modules, Force Structure File

```

'FILE MAR14C.IN;   third test run of March 14 '
'TITLE' 'MAR14C'
'CODE SECTION '    0           EXERCISE BOTH MODULES
'FILE OVERWRITE '  0           DO NOT ALLOW FILE OVERWRITE
'DATA DIRECTORY'  'C:\FACODE\'
'OUTPUT FILE DIRECTORY' 'D:\FM594\OUTEXP\'
'SIMULATION START'   7 1774
'SIMULATION END'     12 1780
'CONFLICT START'     1 1777
'FORCE FILE'         1 'FRCDUM2'      BIG FORCE FILE
'MEI INVENTORY FILE' 2 'DMEIINV2'
'COST FILE'          3 'DUMCOST'
'PROD PROCESS LEAD TIMES' 4 'PDCAPRIL'
'PROD PROCESS MATRIX' 5 'PDCAPRIL'
'BASE MILITARY REQUIREMENTS'           ' 6 'MILAPRY'
'CIVILIAN REQUIREMENTS'                 ' 8 'CIVAPRY'
'Q/K RATIOS AND EOC FRACTIONS'           ' 9 'QKFAPRY'
'SUPPLY SIDE DATA'                     ' 10 'SUPAPRY'
'END OF DATA FILES MARK' 99 'XXXXXX'
'OPTIONAL FILES?' 0
'SPECIAL CIVILIAN FACTOR?' 0
'SPECIAL BASE MILITARY FACTOR?' 0
'LEAD TIME FACTORS' 70 1 999  MULTIPLY FILE PROD'N LEAD TIMES BY 70%
'100% ATTRITION REPLACEMENT?' 'Y' 'N' 'Y' 'N'
'THEATER DEMAND ON INDUSTRY?' 'Y' 'N' 'N' 'N'
'STARTUP COSTS AS WELL AS LOSSES?' 'N' 'N' 'N' 'Y'
'INVENTORY SHARE GROUPS?' 0
'THEATER DATA -- PLAY ONLY THREE THEATERS, DO NOT PLAY THEATER 2 '
'THEATER 1' 1 'Y' 1 1777 7 1 100.0
'THEATER 2' 2 'N' 2 1778 8 1 0.0
'THEATER 3' 3 'Y' 1 1779 9 1 0.0
'THEATER 4' 4 'Y' 1 1780 2 1 0.0
'INVESTMENT DATA'
'EOC EXPANSION PERCENTAGE'           ' 80
'RAMP-UP PERIOD LENGTH (MONTHS)'     ' 6
'NO INVESTMENT?'                     ' 0 (0--DO INVESTMENT)
'INVESTMENT LEAD TIME PERCENTAGE MULTIPLIER' ' 100
'MINIMUM LAG TIME FOR INVESTMENTS (MONTHS)' ' 0
'CONVERGENCE FACTOR PERCENTAGE'      ' 100
'MAXIMUM NUMBER OF ITERATIONS'       ' 500
'MAJOR END ITEM RQMTS IN DOLLARS ' 2 'MAR14C'
'MONTHLY SUPPLY EXPANSION REPORT ' 3 'MAR14C'      for year below
1775
'YEARLY POSTPROCESSOR REPORT WITH COMMAS' ' 15 'MAR14C'
'RANKED SHORTFALL REPORT WITH COMMAS'     ' 17 'MAR14C'
'MILITARY SUPPLY EXPANSION'                ' 29 'MAR14C'
'SUMMARY SPREADSHEET REPORT'              ' 30 'MAR14C'
'END OF FILE INDICATOR' 99 'XXXXXX'
----- comments below
A BIG TEST RUN.
INVESTMENT DATA FILES NOT SPECIFIED. PROGRAM WILL USE DEFAULT-NAMED FILES.

```

2. Sample Control Inputs File for Run Option 1a: Requirements Module Only, Force Structure File

```

'FILE REQST.IN    SAMPLE RUN TO DO REQUIREMENTS MODULE ONLY'
'TITLE'  'REQST'
'CODE SECTION '   1           do requirements module only
'FILE OVERWRITE '   1           do allow file overwrite
'DATA FILE DIRECTORY'  'C:\FACODE\'
'OUTPUT FILE DIRECTORY' 'D:\FM594\OUTEXP\'
'SIM START'      7 1774
'SIM END'       12 1780
'CONFLICT START'  1 1777
'FORCE FILE'      1 'FRCDUM2'
'MEI INVENTORY FILE'  2 'DMEIINV2'
'COST FILE'       3 'DUMCOST'
'PROD PROCESS FILE LEAD TIMES' 4 'PDCAPRIL'
'PROD PROCESS FILE MATRIX'     5 'PDCAPRIL'
'MARK FOR END OF INPUT FILES' 99 'XXXXXX'
'OPTIONAL FILES?'  0
'LEAD TIME FACTORS' 100 1 999
'100% ATTRITION REPLACEMENT?' 'Y' 'Y' 'Y' 'Y'
'THEATER DEMAND ON INDUSTRY?'  'Y' 'N' 'Y' 'Y'
'STARTUP COSTS AS WELL AS LOSSES?' 'Y' 'N' 'Y' 'N'
'INVENTORY SHARE GROUPS?'  1
'MEMBERSHIP IN GROUP '  'Y' 'Y' 'N' 'Y'
'PRIORITY IN GROUP'     1  3  0  2
'THEATER DATA  -- PLAY FOUR THEATERS '
'THEATER 1' 1  'Y'  1 1777 7 10  80.0
'THEATER 2' 2  'Y'  3 1779 8 10   0.0
'THEATER 3' 3  'Y'  1 1778 9 10  20.0
'THEATER 4' 4  'Y'  3 1778 2 10   0.0
'MEI UNIT FILE'         1 'REQST'
'MEI DOLLAR FILE, W. COMMAS'      12 'REQST'
'WEAPON USAGE REPORT '         21 'REQST'
'TOE COMPOSITION FOR MEIs '     45 'REQST1'
10  20
'TOE COMPOSITION FOR MEIs '     45 'REQST2'
79 79          report generated for MEI 79 only
'TOE COMPOSITION FOR MEIs '     45 'REQST3'
135 141
'END OF FILE INDICATOR'      99 'XXXXXX'
-----
CONFLICT MILITARY REQUIREMENTS FILE WILL AUTOMATICALLY BE GENERATED, EVEN
THOUGH NOT REQUESTED.  WILL BE NAMED REQST.CFM, IN DIRECTORY C:\FACODE.

THREE DIFFERENT REQUESTS FOR REPORT 45, TOE COMPOSITION FOR MEIs, FOR
DIFFERENT SETS OF MEIs.  TO AVOID FILE OVERWRITE, SPECIFY DIFFERENT NAMES
FOR EACH OUTPUT REPORT OF THE SAME TYPE.

```

3. Sample Control Inputs File for Run Option 2: Industry-Level Module Only

'FILE ILM001A.IN	TEST OF INDUSTRY-LEVEL MODULE	'
'TITLE		' 'ILM001A'
'CODE SECTION		' 2
'FILE OVERWRITE		' 1
'DATA FILE DIRECTORY		' 'C:\FACODE\'
'OUTPUT FILE DIRECTORY		' 'D:\FM594\OUTEXP\'
'SIM START		' 07 1774
'SIM END		' 12 1780
'BASE MILITARY REQUIREMENTS		' 6 'MILEXPER'
'CONFLICT MILITARY REQUIREMENTS		' 7 'RT001A'
'CIVILIAN REQUIREMENTS		' 8 'CIVAPRY'
'Q/K RATIOS AND EOC FRACTIONS		' 9 'QKFAPRY'
'SUPPLY SIDE DATA		' 10 'SUPEXPER'
'INVESTMENT DISTRIBUTION		' 11 'CMAT'
'INVESTMENT LEAD TIMES		' 12 'GREEN'
'INVESTMENT SECTOR MAPPING		' 13 'CAPIND'
'END OF DATA FILES MARKER		' 99 'XXXXXX'
'OPTIONAL FILES?		' 1
'IMPORT/EXPORT FACTORS		' 3 'IEFDUM'
'SPECIAL CIVILIAN FACTOR?		' 1
2.00 1774 1780		
'SPECIAL BASE MILITARY FACTOR?		' 0
'INVESTMENT DATA'		
'EOC EXPANSION PERCENTAGE		' 100
'RAMP-UP PERIOD LENGTH (MONTHS)		' 0
'NO INVESTMENT?		' 0
'INVESTMENT LEAD TIME PERCENTAGE MULTIPLIER		' 100
'MINIMUM LAG TIME FOR INVESTMENTS (MONTHS)		' 0
'CONVERGENCE FACTOR PERCENTAGE		' 100
'MAXIMUM NUMBER OF ITERATIONS		' 500
'SUMMARY SUPPLY-SIDE SPREADSHEET-READY OUTPUT		' 30 'ILM001A'
'RANKED SHORTFALL REPORT		' 7 'ILM001A'
'SUPPLY AND DEMAND REPORT, BY YEAR		' 4 'ILM001A'
'YEARLY STOCKPILE POSTPROCESSOR REPORT		' 5 'ILM001A'
'QUARTERLY STOCKPILE POSTPROCESSOR REPORT, COMMAS		' 16 'ILM001A'
'YEARLY SUPPLY EXPANSION REPORT, COMMA-DELIMITED		' 19 'ILM001A'
'END OF FILE INDICATOR		' 99 'XXXXXX'

4. Sample Control Inputs File for Run Option 0b: Both Modules, MEI Requirements File

```
'FILE MEITST1.IN      USE MEI REQUIREMENTS FILE BY MEI AND MONTH'
'TITLE'  'MEITST1'
'CODE SECTION '      0      here, do both modules
'FILE OVERWRITE OK?'  1
'DATA DIRECTORY'  'C:\FACODE\'
'OUTPUT FILE DIRECTORY' 'D:\FM594\OUTEXP\'
'SIM START'      7 1774
'SIM END'        12 1780
'CONFLICT START'  6 1779
'MEI INVENTORY FILE ' 2 'NULL'          file specifies zero inventory
'COST FILE'        3 'DUMCOST'
'PROD PROCESS LEAD TIMES' 4 'PDCAPRIL'
'PROD PROCESS MATRIX'  5 'PDCAPRIL'
'BASE MIL RQMTS.'    6 'MILAPRY'
'CIVILIAN FILE'      8 'CIVAPRY'
'SUPPLY SIDE FILE'   10 'SUPAPRY'
'SUPPLY SIDE QKF FILE' 9 'QKFAPRY'
'END OF DATA FILES MARK' 99 'XXXXXX'
'OPTIONAL FILES?'  2
'MEI REQUIREMENTS FILE' 4 'DUMREQ1'
'IMPORT/EXPORT FACTORS' 3 'IEFDUM'
'SPECIAL CIVILIAN FACTOR?' 0
'SPECIAL BASE MILITARY FACTOR?' 0
'LEAD TIME FACTORS' 100 1 999
'THEATER DATA -- PLAY ONE DUMMY THEATER FOR MEI REQUIREMENTS '
'THEATER 1' 1 'Y' 6 1779 14 1 100.0      see note below
'FORM OF MEI REQ.FILE' 1                  see note below
'INVESTMENT DATA'
'EOC EXPANSION % ' 100
'RAMP-UP PERIOD' 6
'NO INVESTMENT?' 1 do no investment. see shortfalls created by MEI dmd
'MEI UNIT FILE' 11 'MEITST1'
'MEI DOLLAR FILE' 12 'MEITST1'
'RANKED SHORTFALL REPORT ' 17 'MEITST1'
'END OF FILE MARK' 99 'XXXXXX'
```

Inventory allocation percentage on theater specification line does not matter, as MEI inventory file specifies no inventory.
In this example, MEI Requirements file has form "1" -- specifies MEI demand for each MEI and month of conflict.

5. Sample Control Inputs File for Run Option 1b: Requirements Module Only, MEI Requirements File

```
'TEST OF REQUIREMENTS MODULE WITH MEI REQUIREMENTS FILE '
'TITLE' 'RQSIMN1'
'CODE SECTION ' 1
'FILE OVERWRITE OK?' 1
'DATA FILE DIRECTORY' 'C:\FACODE\'
'OUTPUT FILE DIRECTORY' 'D:\FM594\OUTEXP\'
'SIMULATION START' 7 1774
'SIMULATION END' 12 1780
'CONFLICT START' 1 1777
'MEI INVENTORY FILE ' 2 'DMEIINV2'
'COST FILE' 3 'DUMCOST'
'PROD PROCESS LEAD TIMES' 4 'PDCAPRIL'
'PROD PROCESS MATRIX' 5 'PDCAPRIL'
'END OF DATA FILES MARKER' 99 'XXXXXX'
'OPTIONAL FILES?' 1
'MEI REQ FILE' 4 'DUMMYRAN'
'LEAD TIME FACTORS' 100 1 999
'THEATER DATA--ONE DUMMY THEATER FOR MEI REQUIREMENTS, 7-MONTH CONFLICT'
'THEATER 1' 1 'Y' 1 1777 7 1 75.0 see note below on inventory alloc.
'FORM OF MEI REQ. FILE' 2 see note below
'MEI REQUIREMENTS DISTRIBUTION' 50. 20. 10. 10. 10. 0. 0.
'MEI UNIT FILE' 1 'RQSIMN1'
'MEI DOLLAR FILE' 2 'RQSIMN1'
'CONFLICT MILITARY FILE' 10 'RQSIMN1' can be explicitly requested
'AGGREGATED MEI REPORT (AGGTAB)' 20 'RQSIMN1' see comment below
'END OF FILE MARK' 99 'XXXXXX'
```

Inventory allocation of 75% means that only 75% of the file values of MEI inventory (in file DMEIINV2.MIN) can be used to help satisfy demand.

In this example, MEI Requirements file has form "2" -- only specifies total MEI demand. Need the "MEI requirements distribution" line to allocate the MEI demand over the seven months of conflict. Same allocation percentages are used for each MEI.

AGGTAB report requires that auxiliary file AGGMAP.DAT, the MEI aggregation mapping file, be present in the data file directory.

III. STRUCTURE AND FORMAT OF THE DATABASE, OPTIONAL, AND AUXILIARY FILES

A. OVERVIEW

This chapter describes the format of the input database, optional, and auxiliary files needed by the FORCEMOB model. The formats are explained in sufficient detail so that a FORCEMOB data preparer or user can prepare and organize the values of the FORCEMOB input variables and arrays in a manner such that FORCEMOB can read them.

The reader is cautioned that the input files must follow rather specific formats. The number of data records and their ordering can depend on the values of certain previously read inputs. The positioning of values on a data record might follow many different formats, depending on the particular input being considered. In Version 3.1, many of the READ statements in the FORCEMOB computer code have been changed to a list-directed (*) format. This allows more format flexibility than previous versions did. However, character inputs in the database files are read in a fixed-column format, and the number of records and their ordering remain extremely important.

FORTTRAN format notation is used frequently. Reading the FORTTRAN code of the various FORCEMOB subroutines that read the input data files might be helpful in clarifying the input formats. Tables II-1 and II-2, in Chapter II, indicate which subroutines read which files.

Section A.1 reviews the information on file selection and file location that appeared in Chapter II. The rest of this chapter is organized on a file by file basis. Each file is discussed in a standard format, as explained in Section A.2. For ease of presentation, the balance of the chapter groups the file discussions as follows:

Section	File(s) Discussed
B	Element database file
C	Database files used by the FORCEMOB Requirements module
D	Database files used by the FORCEMOB Industry-level module

Section	File(s) Discussed
E	Optional files
F	Auxiliary files

(See Chapters I and II for the meaning of this file taxonomy.)

1. File Selection, Naming, and Location

Many files of each type can exist, and the user can select among them at the outset of a FORCEMOB run. A file of a given type must be given an extension, and possibly a name, unique to that type of file. These specifications are as follows:

- The Element database file must have the name ELEMENT.DB (different Element files must reside in different directories)
- Each database and optional file must have a characteristic extension, as shown in Tables II-1 and II-2; the data preparer can choose the name.¹
- The auxiliary Debugging Flags file must have the name DEBUG.FLG.
- The auxiliary file of Major End Item aggregation mappings must have the name AGGMAP.DAT.

As discussed in Chapter II, the Control Inputs file contains the name of a “data file directory.” Except for the auxiliary Debugging Flags file (which is assumed to reside in the directory from which the program is being run), all input files are assumed to reside in the data file directory. For each type of input file, a given FORCEMOB run uses the file that—

- resides in this directory.
- has the required name, the name specified on the Control Inputs file, or the default name (see Tables II-1 and II-2).
- has the appropriate extension.

Except for the Element file and AGGMAP.DAT files, many files of each type can exist in a particular data file directory, and different runs of FORCEMOB can use different files. The Element database file specifies numbers of types of resources and names for these resources. It is anticipated that the Element database file will seldom need change. Moreover, changes in it might well affect all the other input files. For

¹ In accordance with DOS restrictions, names must be 8 or fewer characters long.

consistency, these different file groups should be located in different directories. This is the rationale behind the hard-coding of the Element file name.

2. File Discussion Format

Each input file is discussed in a three-part format. The parts are:

1. A summary of the data in the file, i.e., a general discussion of the type of data the file contains
2. A record and format guide, which consists of a listing of the lines of the file, the variables on these lines, and the FORTRAN READ format associated with the line; this format specifies the columns in which values for these variables must appear
3. Definitions of each symbolic entity in the file

The file discussion appears in a section of text, with three subsections, one corresponding to each item in the above list. The contents of these subsections are as follows.

a. Summary of Data in File Subsection

The first subsection in a section gives an overview of the type of data in the file being described, and, as applicable, special data and file characteristics of which the user should be aware.

b. Record and Format Guide Subsection

This subsection consists of schematic representations of the records in the database. Essentially, this representation is a condensation of the FORTRAN source code of the subroutine that reads the data file, showing the records read, their formats, and the surrounding loop structure. It might be fruitful to read this source code.

The "for" structure in the schematic representation corresponds to the DO loops in the FORCEMOB computer program. (The notation "for I=1,N" means that I assumes, in turn, integer values from 1 through N, inclusive.) Some of the limits on the loops are variable values input from the current file or other database files; some limits are computed in the program. Subsection c defines the variables used for loop indices and limits. Embedded loops occur frequently.

On each particular data line, the data should be located in the columns appropriate for the READ format of that data line (which often is a list-directed format). This format is shown to the right of the data elements read. A familiarity with FORTRAN FORMAT

statement notation is assumed. Many of the READ statements in the FORCEMOB subroutines contain implied DO loops so that multiple array elements can be read from a single data record. The schematic representation shows these implied DO loops; the variables used for indices and limits on these loops are defined in Subsection c.

Comment records (lines) appear throughout the database files, usually at the head of a new section of data. FORCEMOB reads them but ignores their contents. The comment records can contain informative messages that indicate the particular sections of data that follow. Regardless of contents, the comment records themselves *must* appear in the specific places in the data files indicated in the record and format guides. The code of FORCEMOB expects to find and read them. In addition to whole comment records, remarks can be placed on the data lines, in columns beyond the specific READ format for that line.

An ampersand (&) indicates a continuation line: a place where inputs are listed in multiple lines in the schematic representations but should be put on a single input record in the data file. To improve readability, some blank lines appear in the schematic representations, between certain sections of data. However, these blank lines do *not* correspond to blank records in the data files.

For conciseness, no definitions have been put in the Record and Format Guide subsection; these are contained in the next subsection. In preparing data, the user will need to use the definitions to compute values of certain limit variables (i.e., limits on loops); these values then affect the number of records and the number of data values on a record, in accordance with the schematic representation.

c. Definitions of Symbolic Entities Subsection

The third subsection in each section supplies definitions of the "symbolic entities" (encompassing variables, arrays, and symbolic constants) used in the Record and Format Guide subsection. Entities that are inputs for FORCEMOB and whose values are specified in the current database file are marked with an asterisk. The other listed entities are usually indices or limits for the loops, or entities in other database files whose values affect the structure of the current database file. This subsection and the preceding one are best read in conjunction. In the texts of the definitions, \$K means thousands of dollars and \$M, millions of dollars.

Unless explicitly stated in the definitions, symbolic entities follow the standard FORTRAN typing rule: entities whose names begin with the letters I through N are integer; the rest are real.

B. THE ELEMENT DATABASE FILE

1. Summary of Data in File

The Element database file plays a pivotal role in that it identifies the industry sectors, weapon types, and overall dates that underlie any FORCEMOB analysis. The Element file for the current version is only slightly different in format from that of Version 1.

At the beginning of the file appear—

- a “dollar year”; all monetary data must be in that year dollars
- beginning and ending years for the databases. The overall time span of the databases extends from January of the starting year through December of the ending year, inclusive. All scenario dates must fall within this time span.²

The major portion of the file contains names and labels for

- (1) the industries
- (2) Major End Items
- (3) force unit types
- (4) TOE items
- (5) consumption items
- (6) threat items

used in the simulation databases. Also included are data on mappings (correspondence indicators) between—

- Major End Items and Major End Types
- TOE items and Major End Items
- consumption items and Major End Items
- threat items and Major End Items

² **Caution:** The beginning and ending database years must be no more than 14 years apart, i.e., they must cover an overall time span of no more than 15 years. Some reprogramming of FORCEMOB would be necessary to accommodate a longer time span.

Each section of data starts with a comment record in which the user can identify the contents of the section. The major entities specified in this database are as follows:

Industry Names and Labels. The industry names and labels in the file refer to the economic sectors of the input-output model used by FORCEMOB. In this documentation, we use the terms "industry" and "sector" synonymously. The list of industries should span the U.S. economy (at some level of aggregation).

Major End Items. Major End Items (MEIs) are an accounting convention by weapon, military system, personnel, etc., spanning the entire DoD budget. The list of MEIs should include, at some level of aggregation, all the kinds of platforms, munitions, consumables, and support functions for which there might be a demand in the conflict scenarios that the user plans to analyze. The "Major End Item Types" or "Major End Types" represent an aggregation of the MEIs, as specified by the input array MAPMEI in the Element file. The Major End Type aggregation is used when translating MEI demand into industry demand; sections C.4 and C.5, below, explain the meaning of this aggregation.

Force Unit Types. The list of force unit types constitutes the types of combat units which may be played in the conflict simulation of the FORCEMOB Requirements module. The force to be evaluated is composed of some set of these units, arriving (possibly) at several different months in several different theaters. There is a different specification of force unit types for each of the four Services.

TOE Items, Consumption Items, and Threat Items. TOE stands for "Table of Organization and Equipment." This table delineates the types of weapons and systems that compose each of the force unit types. Consumption items are those items demanded by a force unit during combat and which must be supplied to keep it fighting. Threat item usage is not explicitly tied to the number of force units present; instead, an extrinsic requirement is specified in the Force Structure database file. In previous FORCEMOB data sets, most of the precision guided munitions have been put in the threat item category. For each category of item, there is a different list of item types for each Service. (For further information on this taxonomy of items, see the description of the Requirements model in Volume I, Chapter II, of this paper.)

Note that if only the Industry-level module is being exercised (run option 2, as discussed in Chapter II), then FORCEMOB stops reading the Element file after the industry names and labels have been read, and does not read the weapon-based information. If an MEI Requirements file is used (run options 0b and 1b), FORCEMOB

reads the information on force units, TOE items, consumption items, and threat items—
but does not use it further.

2. Record and Format Guide

COMMENT "HEADER LINE AND IDOLYR"	(A)
IDOLYR	(*)
COMMENT "DATABASE YEARS"	(A)
ISYEAR, IEYEAR	(*)
COMMENT "INDUSTRY DATA"	(A)
NIND	(*)
for IND=1,NIND	
IND, INDNAME(IND), INDLABL(IND)	(I3,2X,A30,A15)
COMMENT "MAJOR END ITEM DATA"	(A)
NMEI, NTP	(*)
for IM=1,NMEI	
IM, MEINAME(IM), MEILABL(IM), MAPMEI(IM)	(I3,2X,A30,A15,I3)
COMMENT "FORCE UNIT DATA"	(A)
(NUNIT(ISER), ISER=1, NSER)	(*)
for ISER=1, NSER	
COMMENT "SERVICE NAME"	(A)
for IU=1, NUNIT(ISER)	
IU, UNTNAME(IU, ISER), UNTLABL(IU, ISER), LITEM(IU, ISER)	(I3,2X,A30,A15,I3)
COMMENT "TOE DATA"	(A)
(NTOE(ISER), ISER=1, NSER)	(*)
for ISER=1, NSER	
COMMENT "SERVICE NAME"	(A)
for ITOE=1, NTOE(ISER)	
ITOE, TOENAME(ITOE, ISER), TOELABL(ITOE, ISER), MAPTOE(ITOE, ISER)	(I3,2X,A30,A15,I3)
COMMENT "CONSUMPTION ITEM DATA"	(A)
(NCON(ISER), ISER=1, NSER)	(*)
for ISER=1, NSER	
COMMENT "SERVICE NAME"	(A)
for ICON=1, NCON(ISER)	
ICON, CONNAME(ICON, ISER), CONLABL(ICON, ISER), MAPCON(ICON, ISER)	(I3,2X,A30,A15,I3)
COMMENT "THREAT ITEM DATA"	(A)
(NTRT(ISER), ISER=1, NSER)	(*)
for ISER=1, NSER	
COMMENT "SERVICE NAME"	(A)
for IT=1, NTRT(ISER)	
IT, TRTNAME(IT, ISER), TRTLABL(IT, ISER), MAPTRT(IT, ISER)	(I3,2X,A30,A15,I3)

3. Definitions of Symbolic Entities

*CONLABL(ICON,ISER) = 15-character label for each consumption item (used on reports when space is a problem)

*CONNAME(ICON,ISER) = 30-character name for each consumption item

ICON = Index for type of consumption item within a Service. Ranges from 1 through the input limit variable NCON(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNCON, which must be large enough to encompass all types of consumption items for each Service.

*IDOLYR = Year that dollar data is in.

*IEYEAR = Ending year for the databases.

IM = Index for kind of Major End Item. Ranges from 1 through the input limit variable NMEI. Dimension limit is the symbolic constant LNMEI.

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

*INDLABL(IND) = 15-character label for industry IND (used on reports when space is a problem).

*INDNAME(IND) = 30-character name for industry IND.

ISER = Index for Service: 1 = Army; 2 = Air Force; 3 = Navy; 4 = Marines. The limit variable NSER and the symbolic constant LNSER are fixed at 4, encompassing these Services.

*ISYEAR = Starting year for the databases.

IT = Index for type of threat item within a Service. Ranges from 1 through the input limit variable NTRT(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNTRT, which must be large enough to encompass all types of threat items for each Service.

ITOE = Index for type of TOE item within a Service. Ranges from 1 through the input limit variable NTOE(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNTOE, which must be large enough to encompass all types of TOE items for each Service.

IU = Index for type of force unit within a Service. Ranges from 1 through the input limit variable NUNIT(ISER), where ISER is the Service currently under

consideration. Dimension limit is the symbolic constant LNUNIT, which must be large enough to encompass all types of units for each Service.

*LITEM(IU,ISER) = Index of the TOE item to use as the lead item for unit IU in Service ISER when computing attrition of a unit under the zero replacement rule.

*MAPCON(ICON,ISER) = Index of the Major End Item to which to map the ICON-th type of consumption item in Service ISER.

*MAPMEI(IM) = Index of the Major End Type to which to map the IM-th kind of Major End Item.

*MAPTOE(ITOE,ISER) = Index of the Major End Item to which to map the ITOE-th type of TOE item in Service ISER.

*MAPTRT(IT,ISER) = Index of the Major End Item to which to map the IT-th type of threat item in Service ISER.

*MEILABL(IM) = 15-character label for kind-IM Major End Item (used on reports when space is a problem).

*MEINAME(IM) = 30-character name for kind-IM Major End Item.

*NCON(ISER) = Number of types of consumption items for Service ISER.

*NIND = Number of industries (sectors).

*NMEI = Number of kinds of Major End Items.

*NTOE(ISER) = Number of types of TOE items for Service ISER.

*NTRT(ISER) = Number of types of threat items for Service ISER.

*NTYP = Number of Major End Types (this is generally different from NMEI, the number of kinds of Major End Items; see MAPMEI).

*NUNIT(ISER) = Number of unit types for Service ISER.

*TOELABL(ITOE,ISER) = 15-character label for the ITOE-th type of TOE item in Service ISER. (Used on reports when space is a problem.)

*TOENAME(ITOE,ISER) = 30-character name for the ITOE-th type of TOE item in Service ISER.

*TRTLABL(IT,ISER) = 15-character label for the IT-th type of threat item in Service ISER. (Used on reports when space is a problem.)

*TRTNAME(IT,ISER) = 30-character name for the IT-th type of threat item in Service ISER.

*UNTLABL(IU,ISER) = 15-character label for the IU-th type of force unit in Service ISER. (Used on reports when space is a problem.)

*UNTNAME(IU,ISER) = 30-character name for the IU-th type of force unit in Service ISER.

C. INPUT DATABASE FILES—REQUIREMENTS MODULE

1. Force Structure Database File

a. Summary of Data in File

The Force Structure database remains as it was in Version 1.0. It contains all of the data used for a conflict simulation in FORCEMOB. These data include complete tables of organization and equipment (TOEs), consumption rates for force units, threat item types and availability, and unit deployment information. For each theater ITHR, values are given by relative month 1 through NMON2(ITHR) (i.e., by month of the conflict period in theater ITHR). The user can then specify on the Control Inputs file the month (within the simulation period) in which the conflict period begins, for each theater. The file also contains threat factors for each Service and theater. Array limits are first given, along with the number of theaters, the global threat requirements, and the threat allocation among theaters. Then follows a complete set of data for each theater specified. Between each "block" of data there is a header record with values to be used as a check on indices. For example, each block of 12 months of unit delivery profiles for a Service will have a record preceding it with the theater number, Service number, and "year" number.

The user should be aware of the following caveat. Most requirements for TOE, consumption, and threat items are expressed in terms of numbers of items required. For some types of these items, however, it might be appropriate to specify a *dollar* amount of requirement (e.g., one type-*x* unit contains *y* thousand dollars worth of TOE item *z*). There is no explicit variable in the file for the year in which to express these dollar values, and the FORCEMOB code performs no checks on this dollar year. The user is responsible for ensuring that this dollar year is the same as IDOLYR, as specified in the Element database file.

b. Record and Format Guide

```

(NUNITF(ISER), ISER=1, NSER), (NTOEF(ISER), ISER=1, NSER),
& (NCONF(ISER), ISER=1, NSER), (NTRTF(ISER), ISER=1, NSER),
& NTHR2, (NMON2(ITHR), ITHR=1, NTHR2), NLVL (*)

for IT=1, MAXTRT
  IT, (GLOTHRT(IT, ISER), ISER=1, NSER) (*)

for ISER=1, NSER
  for IT=1, NTRT(ISER)
    IT, (CONTHRT(IT, ISER, ITHR), ITHR=1, NTHR2) (*)

for ITHR=1, NTHR2 (Note: long loop on theater)
  ITHR, THRNAME(ITHR) (I3, 2X, A15)
  for IMN=1, NBLMON
    for ISER=1, NSER
      ITHR, ISER, IMN (*)
      for IU=1, NUNIT(ISER)
        IU, (UNITS(IMON, IU, ISER, ITHR), IMON=IPST, IPEND) (*)

  for ISER=1, NSER
    ITHR, ISER
    for ITOE=1, NTOE(ISER)
      ITOE, (ATTRATE(IL, ITOE, ISER, ITHR), IL=1, NLVL) (*)

  for IMN=1, NBLMON
    ITHR, IMN (*)
    for ISER=1, NSER
      ISER, (IUNTATT(IMON, ISER, ITHR), IMON=IPST, IPEND) (*)

  for ISER=1, NSER
    for IBU=1, NBLUNIT
      ITHR, ISER, IUST (*)
      for ITOE=1, NTOE(ISER)
        ITOE, (UNITTOE(ITOE, IU, ISER, ITHR), IU=IUST, IUEND) (*)

  for ISER=1, NSER
    for IU=1, NUNIT(ISER)
      ITHR, ISER, IU (*)
      for ICON=1, NCON(ISER)
        ICON, (CONRATE(IL, ICON, IU, ISER, ITHR), IL=1, NLVL) (*)

  for IMN=1, NBLMON
    ITHR, IMN (*)
    for ISER=1, NSER
      ISER, (IUNTCON(IMON, ISER, ITHR), IMON=IPST, IPEND) (*)

  for IMN=1, NBLMON
    for ISER=1, NSER
      ITHR, ISER, IMN (*)
      for IT=1, NTRT(ISER)
        IT, (PERTHRT(IMON, IT, ISER, ITHR), IMON=IPST, IPEND) (*)

```

c. Definitions of Symbolic Entities

*ATTRATE(IL,ITOE,ISER,ITHR) = Monthly attrition rate for TOE item ITOE for Service ISER in theater ITHR, where IL is the intensity level. Data are in fractions of items lost to the force. Values range from 0.0 to 1.0. For those theaters where the "zero attrition replacement rule" is in effect, only the attrition rate for the lead TOE item in a unit (variable LITEM, in the Element database file) is used.

*CONRATE(IL,ICON,IU,ISER,ITHR) = Monthly consumption rate for consumption item ICON in force unit type IU in Service ISER in theater ITHR, where IL is the intensity level. Data are in items consumed by the force.

*CONTHRT(IT,ISER,ITHR) = Proportion of requirement for type-IT threat items in Service ISER to be used for the conflict in theater ITHR. For each particular threat-item/Service combination, the sum over theaters of CONTHRT will normally equal 1.0 but is not required to; if it does not, the overall threat item requirement will differ from the input GLOTHRT.

*GLOTHRT(IT,ISER) = Global requirement of threat item IT in Service ISER. Note that for a given Service ISER, for values of IT greater than NTRT(ISER) but less than or equal to MAXTRT, (dummy) values of GLOTHRT(IT,ISER) must appear on the file, but are not used.

IBU = Index for groups of unit types. IBU ranges from 1 through NBLUNIT, inclusive.

ICON = Index for type of consumption item within a Service. Ranges from 1 through the input limit variable NCON(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNCON, which must be large enough to encompass all types of consumption items for each Service.

IL = Index for combat intensity level. Used with certain variables governing attrition and consumption. Ranges from 1 through the input limit variable NLVL. Dimension limit is symbolic constant LNLVL.

IMN = Index for a specific 12-month period in a given theater. IMN ranges from 1 through NBLMON, inclusive.

IMON = Index for month of combat within a specific theater, commencing from the start date of combat in that theater. Here, IMON ranges from 1 through 12*NBLMON.

IPEND = Defined by $IPEND = IMN * 12$, or $IPEND = IPST + 11$, for a specific 12-month period IMN in the range 1,...,NBLMON, inclusive. Indexes ending month of this period.

IPST = Defined by $IPST = (IMN-1)*12 + 1$, for a specific 12-month period IMN in the range 1,...,NBLMON, inclusive. Indexes starting month of this period.

ISER = Index for Service: 1 = Army; 2 = Air Force; 3 = Navy; 4 = Marines. The limit variable NSER and the symbolic constant LNSER are fixed at 4, encompassing these Services.

IT = Index for type of threat item within a Service. Ranges from 1 through the input limit variable NTRT(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNTRT, which must be large enough to encompass all types of threat items for each Service.

ITHR = Index of theater. Ranges from 1 through the limit variable NTHR. Dimension limit is the symbolic constant LNTHR.

ITOE = Index for type of TOE item within a Service. Ranges from 1 through the input limit variable NTOE(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNTOE, which must be large enough to encompass all types of TOE items for each Service.

IU = Index for type of force unit within a Service. Ranges from 1 through the input limit variable NUNIT(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNUNIT, which must be large enough to encompass all types of units for each Service.

IUEND = Defined by $IUEND = \min\{IUST+9, NUNIT(ISER)\}$, for the IBU-th group of ten unit types in Service ISER, for IBU in the range 1,...,NBLUNIT, inclusive. IUEND indexes ending unit type in group.

*IUNTATT(IMON,ISER,ITHR) = Index of attrition intensity level to be used from ATTRATE for month IMON in Service ISER in theater ITHR. (Note: for each theater/Service combination, $12*NBLMON$ entries are required in the file, but entries beyond $NMON2(ITHR)$ are not used.)

*IUNTCON(IMON,ISER,ITHR) = Index of consumption intensity level to be used from CONRATE for month IMON in Service ISER in theater ITHR. (Note: for each theater/Service combination, $12*NBLMON$ entries are required in the file, but entries beyond $NMON2(ITHR)$ are not used.)

IUST = Defined by $IUST = (IBU-1)*10 + 1$, for the IBU-th group of ten unit types (in a given Service), for IBU in the range 1,...,NBLUNIT, inclusive. IUST indexes starting unit type in group.

MAXTRT = The maximum number of types of threat items in a Service; defined as the maximum value, over the Services ISER, of NTRT(ISER).

NBLMON = Defined as the smallest integer greater than or equal to $NMON(ITHR)/12$, for a specific theater ITHR. Number of years (rounded upward to an integer) of conflict in theater ITHR. Redefined for each theater, in turn.

NBLUNIT = Defined as the smallest integer greater than or equal to $NUNIT(ISER)/10$, for a specific Service ISER. Number of groups of 10 unit types for that Service.

NCON(ISER) = Number of types of consumption items for Service ISER.

NCONF(ISER) = Number of types of consumption items for service ISER on the file under consideration. Value must equal NCON(ISER), as specified in the Element file.

*NLVL = Number of combat intensity levels.

*NMON2(ITHR) = Initial specification of number of months of conflict in theater ITHR (upper limit on value that can be set in the Control Inputs file).

NSER = Number of Services. Value is fixed in code at 4, comprising 1—Army; 2—Air Force; 3—Navy; 4—Marines. See also ISER.

*NTHR2 = Initial number of theaters being played.

NTOE(ISER) = Number of types of TOE items for Service ISER.

NTOEF(ISER) = Number of types of TOE items for service ISER on the file under consideration. Value must equal NTOE(ISER), as specified in the Element file.

NTRT(ISER) = Number of types of threat items for Service ISER.

NTRTF(ISER) = Number of types of threat items for service ISER on the file under consideration. Value must equal NTRT(ISER), as specified in the Element file.

NUNIT(ISER) = Number of unit types for Service ISER.

NUNITF(ISER) = Number of unit types for service ISER on the file under consideration. Value must equal NUNIT(ISER), as specified in the Element file.

*PERTHRT(IMON,IT,ISER,ITHR) = Distributes conflict threat percentage (CONTHRT) over months (IMON) of conflict in theater ITHR, for the IT-th type of threat item in Service ISER. For each particular threat-item/Service/theater combination, the sum over months (1 through NMON2(ITHR)) of PERTHRT will normally equal 1.0 but is not required to; if it does not, the overall threat item requirement will in general differ from the input GLOTHRT. (Note: for each threat-item/Service/theater combination, $12 \times NBLMON$ entries are required in the file, but entries beyond NMON2(ITHR) are not used.)

*THRNAME(ITHR) = 15-character label for theater ITHR.

*UNITS(IMON,IU,ISER,ITHR) = Number of force units of type IU in Service ISER arriving in theater ITHR in month IMON of the conflict in theater ITHR; i.e., the monthly delivery profile of force units. (Note: for each theater/Service/unit-type combination, 12*NBLMON entries are required in the file, but entries beyond NMON2(ITHR) are not used.)

*UNITTOE(ITOE,IU,ISER,ITHR) = Number of TOE items of type ITOE in one force unit of type IU in Service ISER in theater ITHR.

2. Major End Item Inventory File

a. Summary of Data in File

This file contains initial amounts of inventory for the Major End Items. This inventory can be used to offset MEI demand that arises from the conflict scenario. In previous versions of FORCEMOB, the inventory amounts were either computed quantities or were inputs in the Base Military Requirements database file. In Version 3, we have relocated this information to its own file. The inventory is expressed in dollar terms, and MEI demand, which is also expressed in dollar terms, is ameliorated by the inventory on a dollar-for-dollar basis (see the discussion of the Cost database file, below). Not all MEIs need appear in the file—those that do not are given an inventory level of zero. If the user decides not to model any inventory at all, the Major End Item Inventory file must still appear—but it can specify all zero values.

As discussed in Chapter II, the Control Inputs file contains factors that allocate the inventory among the theaters played. FORCEMOB does not require that these allocation factors sum to 100 percent; if they don't, the effective amount of inventory will be different from the value appearing in the MEI inventory file. An inventory allocation file (Optional File 5) can also be used to allocate inventory among theaters. A separate demand amelioration procedure is performed for each theater—but after that, inventory sharing, if specified on the Control Inputs file, allows some inventory that had been earmarked for one theater to be used to offset demand in other theaters.

A given Major End Item might correspond to an aggregation of TOE, consumption, and/or threat items. In determining a total dollar value for the inventory for that MEI, one might want to multiply each corresponding TOE, consumption, or threat

item inventory by an appropriate cost (see the discussion of the Cost database file) and sum the results.

b. Record and Format Guide

NMEIF, IINVYR

(*)

for I=1,...(until end of file)
IM, CUMINV(IM)

(*)

c. Definitions of Symbolic Entities

*CUMINV(IM) = Cumulative inventory for Major End Item kind IM at start of conflict. Values in \$K.

*IINVYR = Dollar year in which MEI inventory amounts are expressed. Must be the same as IDOLYR, as specified in the Element database file.

IM = Index for kind of Major End Item being considered on the current data line.

NMEI = Number of kinds of Major End Items.

NMEIF = Number of kinds of Major End Items. Value must equal NMEI, as specified in the Element file. (Used as a check.)

3. Cost Database File

a. Summary of Data in File

The Cost database file has the same format as in Version 1.0. It contains cost data for the Major End Items, TOE items, consumption items, and threat items. It also contains "substitution factors," as discussed below, and informative units of measure for the TOE, consumption, and threat items (these units appear on some output reports, but do not affect the FORCEMOB calculations). Item costs generally correspond to the common conception of costs and must be expressed in the same dollar year as IDOLYR, specified in the Element file. But if the corresponding TOE, consumption, or threat item requirement is expressed in dollar terms (see the discussion of the Force Structure file), then unity (i.e., 1.0) is a reasonable value for the cost.

The following points should be noted about the MEI prices (variable PRICE(IM)), which appear in the first major section of the file:

- The prices are needed only for output reports; they do not affect the calculations of FORCEMOB.

- Even so, a price should appear for every MEI.
- Avoid prices of zero (although the program will still run in that case).
- If an MEI Requirements file is used (Optional File 4; run options 0b or 1b of Chapter II), then the data relating to TOE, consumption, and threat items are not relevant, are not needed, and are not read. The MEI prices *are* needed, but only for output reports. (The MEI Requirements file expresses the requirements in dollar terms.)

Some discussion of the substitution factors is in order. Let us consider TOE items. For a given TOE item, say item type ITOE in Service ISER, FORCEMOB computes an initial amount required and then multiplies this by the term $\text{TOECOST}(\text{ITOE}, \text{ISER}) * \text{TOESUB}(\text{ITOE}, \text{ISER})$ to obtain a dollar amount required.³ This dollar amount becomes part of the of the MEI demand for the MEI to which item ITOE corresponds.⁴ Here, TOECOST is the cost and TOESUB, the substitution factor. Generally, the “initial amount” is a number of items, but possibly, the initial amount is a dollar amount and the value of TOECOST is unity (see above). In either case, the value of TOESUB can reasonably be unity, unless one deliberately wishes to discount or otherwise affect the relative value of certain TOE items. Similar considerations apply to consumption and threat item substitution factors (variables CONSUB and TRTSUB).

Each Major End Item represents an aggregate of certain TOE, consumption, and/or threat items. Suppose that TOE item ITOE in Service ISER is associated with Major End Item IM (i.e., $\text{IM} = \text{MAPTOE}(\text{ITOE}, \text{ISER})$; the array MAPTOE is set in the Element database file). Then the quantity

$$w = \text{TOECOST}(\text{ITOE}, \text{ISER}) * \text{TOESUB}(\text{ITOE}, \text{ISER}) / \text{PRICE}(\text{IM})$$

can be considered as an “equivalency factor” that relates TOE item quantities to the corresponding Major End Item. That is, FORCEMOB in effect considers a requirement for x TOE items to be equivalent to a requirement for the quantity wx of the corresponding MEI. If the TOESUB value is 1.0, then tradeoffs between the TOE item and the corresponding MEI are made on a dollar for dollar basis. A similar interpretation applies to consumption and threat items.

³ For TOE items only, and only in computing unit startup requirements, the input factor $\text{PROFACT}(\text{ITOE}, \text{ISER})$ also appears in this product.

⁴ The demand for a given MEI—expressed in dollar terms—is the sum of the dollar amounts required of all the TOE, consumption, and threat items that correspond to that MEI.

b. Record and Format Guide

```

NMEIF, IPYEAR, (NTOEF (ISER), ISER=1, NSER),
& (NCONF (ISER), ISER=1, NSER), (NTRTF (ISER), ISER=1, NSER)          (*)

COMMENT      "MEI PRICE"                                             (A)
┌for IM=1, NMEI
│  IM, PRICE (IM)                                                    (*)
└──────────────────────────────────────────────────────────────────────────

┌for ISER=1, NSER
│  COMMENT      "TOE COST DATA FOR SERVICE ____"                    (A)
│  ┌for ITOE=1, NTOE (ISER)
│  │  ITOE, TOEUM (ITOE, ISER), TOECOST (ITOE, ISER),
│  │  &      TOESUB (ITOE, ISER), PROFAC (ITOE, ISER)                (I3, 2X, A5, 3F12.2)
│  └──────────────────────────────────────────────────────────────────
└──────────────────────────────────────────────────────────────────────────

┌for ISER=1, NSER
│  COMMENT      "CONSUMPTION COST DATA FOR SERVICE ____"            (A)
│  ┌for ICON=1, NCON (ISER)
│  │  ICON, CONUM (ICON, ISER), CONCOST (ICON, ISER), CONSUB (ICON, ISER)  (I3, 2X, A5, 3F12.2)
│  └──────────────────────────────────────────────────────────────────
└──────────────────────────────────────────────────────────────────────────

┌for ISER=1, NSER
│  COMMENT      "THREAT COST DATA FOR SERVICE ____"                (A)
│  ┌for IT=1, NTRT (ISER)
│  │  IT, TRTUM (IT, ISER), TRTCOST (IT, ISER), TRTSUB (IT, ISER)        (I3, 2X, A5, 3F12.2)
│  └──────────────────────────────────────────────────────────────────
└──────────────────────────────────────────────────────────────────────────

```

c. Definitions of Symbolic Entities

*CONCOST(ICON,ISER) = Unit cost (in \$K) for the ICON-th type of consumption item in Service ISER.

*CONSUB(ICON,ISER) = Substitution factor applied to mapping from consumption item to Major End Item.

*CONUM(ICON,ISER) = 5-character label for unit of measure for consumption item for Service ISER

ICON = Index for type of consumption item within a Service. Ranges from 1 through the input limit variable NCON(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNCON, which must be large enough to encompass all types of consumption items for each Service.

IM = Index for kind of Major End Item. Ranges from 1 through the input limit variable NMEI. Dimension limit is the symbolic constant LNMEI.

*IPYEAR = Price index year for the price and cost data. All prices must be for the same year, and this year must equal the variable IDOLYR, as specified in the Element file.

ISER = Index for Service: 1 = Army; 2 = Air Force; 3 = Navy; 4 = Marines. The limit variable NSER and the symbolic constant LNSER are fixed at 4, encompassing these Services.

IT = Index for type of threat item within a Service. Ranges from 1 through the input limit variable NTRT(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNTRT, which must be large enough to encompass all types of threat items for each Service.

ITOE = Index for type of TOE item within a Service. Ranges from 1 through the input limit variable NTOE(ISER), where ISER is the Service currently under consideration. Dimension limit is the symbolic constant LNTOE, which must be large enough to encompass all types of TOE items for each Service.

NCON(ISER) = Number of types of consumption items for Service ISER.

NCONF(ISER) = Number of types of consumption items for service ISER on the file under consideration. Value must equal NCON(ISER), as specified in the Element file.

NMEI = Number of kinds of Major End Items.

NMEIF = Number of kinds of Major End Items on the file under consideration. Value must equal NMEI, as specified in the Element file.

NSER = Number of Services. Value is fixed in code at 4, comprising 1—Army; 2—Air Force; 3—Navy; 4—Marines. See also ISER.

NTOE(ISER) = Number of types of TOE items for Service ISER.

NTOEF(ISER) = Number of types of TOE items for service ISER in the file under consideration. Value must equal NTOE(ISER), as specified in the Element file.

NTRT(ISER) = Number of types of threat items for Service ISER.

NTRTF(ISER) = Number of types of threat items for service ISER on the file under consideration. Value must equal NTRT(ISER), as specified in the Element file.

*PRICE(IM) = Unit price (in \$K) for a kind-IM Major End Item.

*PROFACT(ITOE,ISER) = Procurement factor for the ITOE-th type of TOE item in Service ISER. Used to adjust the item's dollar cost when the item has associated procurement costs in addition to its base price (TOECOST is multiplied by PROFACT). Used only in computing unit startup requirement costs.

*TOECOST(ITOE,ISER) = Unit cost for the ITOE-th type of TOE item in Service ISER. (\$K)

*TOESUB(ITOE,ISER) = Substitution factor applied to mapping from TOE item to Major End Item.

*TOEUM(ITOE,ISER) = 5-character label for unit of measure for the ITOE-th type of TOE item in Service ISER.

*TRTCOST(IT,ISER) = Unit cost for the IT-th type of threat item in Service ISER. (\$K)

*TRTSUB(IT,ISER) = Substitution factor applied to mapping from threat item to Major End Item.

*TRTUM(IT,ISER) = 5-character label for unit of measure for the IT-th type of threat item in Service ISER.

4. Production Process Lead Times File

a. Summary of Data in File

This file contains lead times, in months, required to produce each Major End Item. In Versions 1 and 2 of FORCEMOB, this information appears in the same file as the matrix information (Section 5, below), in binary form (the formats for Versions 1 and 2 are somewhat different from each other). In Version 3, the lead times appear (in ASCII format) in a separate (short) file.

Note that the lead times are organized by Major End Type, not Major End Item. FORCEMOB assumes and requires that different MEIs with the same associated Major End Type (variable MAPMEI, in the Element file) have the same production lead time. (If there is a disparity in the lead times of MEIs associated with the same Major End Type, use the longest one.)

As explained in Chapter II, the Control Inputs file has factors that can be used to adjust the lead time values in the file. (The file value is adjusted by the factors to determine the value that FORCEMOB will use.) One common interpretation is that the file has peacetime lead times, and the percentage factor in the Control Inputs file can reduce these lead times, in concordance with a mobilization scenario. (The factor can also exceed 100 percent, to examine the effect of longer lead times.)

b. Record and Format Guide

NTYPF	(*)
for ITYP=1, NTYP	
ITYP, LDTIMB(ITYP)	(*)

c. Definitions of Symbolic Entities

ITYP = Index of Major End Type. Ranges from 1 through the input limit variable NTYP. Dimension limit is the symbolic constant LNTYP.

NTYP = Number of Major End Types (this is generally different from NMEI, the number of kinds of Major End Items; see MAPMEI, in the Element file).

NTYPF = Number of Major End Types in the data file under consideration. Value must equal NTYP, as specified in the Element file.

*LDTIMB(ITYP) = Input lead time (months) to produce Major End Type ITYP.

5. Production Process Matrix File

a. Summary of Data in File

The Production Process Matrix data provide the link between demands for weapons and demand on industry. The entries give, for each MEI, the dollar amount of industry contributions, ranging over all the industries, required to make one dollar's worth of that MEI. That is, the data in the file contain the distribution of a dollar of demand for each Major End Type (see below) across each industry, in total requirements terms. The methodology and data behind the construction of this file have been explained elsewhere ([1], [3], [4], and in Volume I, Chapter II, of the current paper). In the current version of FORCEMOB (and also in Version 2) the demand on industry is assumed to occur uniformly over the lead time of the Major End Type.

This section describes the format of the file, but we recommend that the file not be prepared by hand. In IDA's work with FORCEMOB, we have used a special preprocessor program that multiplies the Defense Translator data by the Leontief inverse matrix [1, 4] and writes out the entries of the resulting matrix in the format shown here. (For more information about the Defense Translator, see Volume I, Chapter II, section A.4 of this paper; also see Frazier, Campbell, and Cheslow [12].)

As with the production process lead times, the data are organized by Major End Type, not Major End Item. The very definition of Major End Type is based in the

Defense Translator vectors: two MEIs have the same associated Major End Type if and only if they have the same Defense Translator vector. The file has one data line, with one data value on it, for each combination of industry and Major End Type. For the current data sets, this adds up to about 50,000 lines—but still, the file is considerably smaller than the production process file used in Version 1 of FORCEMOB. It is somewhat larger than the binary file of Version 2.2, but for code portability and ease of file management, we wished to get away from using binary files.

b. Record and Format Guide

```

NTYPF, NINDF                                     ( * )
└─for ITYP=1, NTYP
  └─for IND=1, NIND
    ITYP, IND, PDSCST(IND, ITYP)                 ( * )
  └─
└─

```

c. Definitions of Symbolic Entities

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

ITYP = Index of Major End Type. Ranges from 1 through the input limit variable NTYP. Dimension limit is the symbolic constant LNTYP.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element file.

NTYP = Number of Major End Types (this is generally different from NMEI, the number of kinds of Major End Items; see MAPMEI).

NTYPF = Number of Major End Types in the data file under consideration. Value must equal NTYP, as specified in the Element file.

*PDSCST(IND, ITYP) = Dollar amount of total requirements demand on industry IND necessary to produce one dollar's worth of Major End Type ITYP.

D. INPUT DATABASE FILES—INDUSTRY-LEVEL MODULE

1. Base Military Requirements Database File

a. Summary of Data in File

Base military requirements are the total requirements impacts upon the U.S. economy from peacetime military spending, including all direct and indirect impacts from this spending. In the past, such data was supplied by DoD on an MEI basis, and the file contained this and initial MEI inventory data for weapons, systems, etc. In the MEI form, the file was then passed through FORCEMOB to derive the total requirements impacts on industry. More recently, defense expenditure estimates have been derived directly from aggregate DoD budget estimates and then passed through a commercial model, such as INFORUM's LIFT model [7, 8], to estimate total requirements impacts for each industry sector. FORCEMOB can then simply read these total requirements impacts, for each industry, rather than computing them.

Versions 1 and 2 of FORCEMOB allowed the base military requirements to be input either by MEI or industry. But since the MEI format was not being used, and since we wanted to facilitate partitioning FORCEMOB into separate Requirements and Industry-level modules, Version 3.1 only allows the "by industry" format. The "by MEI" option might be reinstated in a future code version.

Version 3 accepts industry demand data on a yearly, rather than monthly, basis. The yearly demand is divided by 12 to obtain a monthly value.

FORCEMOB has a number of optional data factors that can be used to modify the base military demand. These factors include:

1. The conflict and post-conflict factors, CFACT(1) and PCFACT(1), which can affect base military demand for each month of the conflict and post-conflict periods, respectively (the same value is used for each industry). These factors are part of the Base Military Requirements file.
2. The optional Base Military Factors file (section E.1, below)
3. A factor that can be specified on the Control Inputs file (as described in Chapter II) to operate during certain (specified) calendar years, for all industries.

If factor sets 2 and 3 both might apply to a given month, the value in set 3 takes precedence over the value in set 2. The conflict and post-conflict factors always apply for

the appropriate months, along with any other factors. Of course, if no factor sets apply, unity (1.0) is the *de facto* factor value. The monthly value that FORCEMOB uses is given by the product

$$\begin{aligned} & (\text{yearly file value}/12) \times (\text{conflict or post-conflict factor if appropriate}) \\ & \times (\text{set 2 or set 3 factor, if specified}). \end{aligned}$$

b. Record and Format Guide

ISYBAS, IEYBAS, INTBAS, IBASYR, NINDF, IBASTYPE	(*)
COMMENT "CONFLICT AND POST-CONFLICT FACTORS"	(A)
PCFACT(1), CFACT(1)	(*)
COMMENT "BASE MILITARY REQUIREMENTS"	(A)
for IND=1, NIND	
IND, (BASRQY(IYR, IND), IYR=ISYRB, IEYRB)	(*)

c. Definitions of Symbolic Entities

*BASRQY(IYR,IND) = Base military demands (in \$M) on industry IND in year IYR.

*CFACT(1) = Factor to indicate the fraction of base military requirements to be met during the conflict period. The factor will be applied during all months of the conflict period to obtain the base military requirements values used then. Although CFACT is declared as an array, only the first element is used, and it applies to all industries.

*IBASTYPE = Way in which base military demand is specified: =0, by MEI (not currently available); =1, by industry. (In current code version, IBASTYPE must equal 1.)

*IBASYR = Index year for the base military requirements data. All such data must be for the same year, and this year must be the same as IDOLYR, as specified in the Element file.

*IEYBAS = Ending year of data for base military requirements. Must equal IEYEAR, as specified in the Element database file.

IEYRB = Index for ending year of base military data, relative to overall starting year of databases. Defined by $IEYRB = IEYBAS - ISYEAR + 1$, where ISYEAR is as read in from the Element database. (Since IEYBAS is required to equal IEYEAR, IEYRB is equal to the number of years of data in all databases.)

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

*INTBAS = Number of intervals within a year across which annual base military requirements are to be distributed (assumed to be 12 months). This variable is not currently used explicitly.

*ISYBAS = Starting year of data for base military requirements. Must equal ISYEAR, as specified in the Element database file.

ISYRB = Index for starting year of base military data, relative to overall starting year of databases. Defined by $ISYRB = ISYBAS - ISYEAR + 1$, where ISYEAR is as read in from the Element database. (Since ISYBAS is required to equal ISYEAR, ISYRB is equal to 1.)

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element file.

*PCFACT(1) = Fraction to indicate the level of base military requirements to be met during the post-conflict period, i.e., the period from the end of the battle to the end of the simulation. The factor will be applied during all months of the post-conflict period to obtain the base military requirements used then. Although PCFACT is declared as an array, only the first element is used, and it applies to all industries.

2. Conflict Military Requirements Database File

a. Summary of Data in File

The Conflict Military Requirements database file gives the demands on industry, in total requirements terms, that is generated by the Major End Item demand associated with a conflict. Values are by month (not year) and industry. An input to the Industry-level module, this file is an *output* of the Requirements module—and will normally be generated by a run of the Requirements module. We show its format here primarily for informative purposes. To ensure consistency of results, we recommend that the user *not* try to edit this file or construct one by hand. If the user wishes to examine the effect of a change in conflict requirements, the appropriate data files for the *Requirements* module should be modified, the Requirements module rerun, and the resulting Conflict Military Requirements file used.

b. Record and Format Guide

```
JSYEAR, JEYEAR, INTCFM, ICFMYR, NINDF          (*)
JBEG(1), JBEG(2), JEND(1), JEND(2)            (*)
ICBEG(1), ICBEG(2), ICEND(1), ICEND(2)        (*)

  for IYR=ISYRD, IEYRD
    for IND=1, NIND
      IND, DELTA(IP, IND), IP=ISPER, IEPER      (*)
```

c. Definitions of Symbolic Entities

- *DELTA(IP,IND) = Delta or conflict military requirements (demands) on industry IND during period IP. Values in \$M.
- *ICBEG(I) = Beginning period for overall conflict. I=1, month (1-12); I=2, year (4 digits).
- *ICEND(I) = Month (ICEND(1)) and year (ICEND(2)) that the conflict ends.
- *ICFMYR = Dollar year in which conflict military requirement amounts are expressed. Must be the same as IDOLYR, as specified in the Element database file.
- IEPER = Defined by $IEPER = IYR * 12$, for a specific 12-month period IYR. IEPER indexes ending month of this period. IYR value ranges over some appropriate span of years.
- IEYRD = Index for ending year of conflict requirements data, relative to the overall starting year of the databases. Defined by $IEYRD = IEND(2) - ISYEAR + 1$, where ISYEAR is as read in from the Element database. IEND(2) is the ending year of the scenario, as specified in the Control Inputs file; this year is also the ending year of conflict requirements data. (Data for all months of this year must appear on the file; use zero values for months beyond the last month of the scenario.)
- IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.
- *INTCFM = (Dummy input; value must appear but is not currently used.)
- IP = Index of month or period. Ranges depend on specific use. Symbolic constant LNPER is an upper bound on the number of values, and is used as a dimension limit on variables that have an IP dimension.
- ISPER = Defined by $ISPER = 1 + (IYR - 1) * 12$, for a specific 12-month period IYR. ISPER indexes starting month of this period. IYR value ranges over some appropriate span of years.

ISYRD = Index for starting year of conflict requirements data, relative to the overall starting year of the databases. Defined by $ISYRD = IBEG(2) - ISYEAR + 1$, where ISYEAR is as read in from the Element database. IBEG(2) is the starting year of the scenario, as specified in the Control Inputs file; this year is also the starting year of conflict requirements data. (Data for all months of this year must appear on the file; use zero values for months preceding the starting month of the scenario.)

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

*JBEG(I) = Variable to ensure consistency of the Conflict Military Requirements file with the rest of the data. Values for JBEG(1) and JBEG(2) must equal the values of IBEG(1) and IBEG(2), the starting month and year of the scenario period, as specified in the Control Inputs file.

*JEND(I) = Variable to ensure consistency of the Conflict Military Requirements file with the rest of the data. Values for JEND(1) and JEND(2) must equal the values of IEND(1) and IEND(2), the ending month and year of the scenario period, as specified in the Control Inputs file.

*JEYEAR = Variable to ensure consistency of the Conflict Military Requirements file with the rest of the data. Value must equal the value of IEYEAR (the ending year for the databases) as specified in the Element database file.

*JSYEAR = Variable to ensure consistency of the Conflict Military Requirements file with the rest of the data. Value must equal the value of ISYEAR (the starting year for the databases) as specified in the Element database file.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element file.

3. Civilian Consumption Requirements Database File

a. Summary of Data in File

The Civilian Consumption Requirements database file contains data for total requirements, by year and industry, associated with civilian consumption. By civilian consumption we mean all non-military civilian demands on the economy, excluding imports and exports, but including investment (i.e., peacetime construction and producer

durables). This file is used to simulate the level of civilian demands on the economy during a conflict.

In previous FORCEMOB versions, there was a separate data value for each month and industry. Version 3 accepts industry demand data on a yearly, rather than monthly, basis. The yearly demand is divided by 12 to obtain a monthly value. As with the base military demand, FORCEMOB has a number of optional data factors that can be used to modify the civilian demand. The monthly value that FORCEMOB uses is given by

$$(\text{yearly file value}) \times (\text{appropriate factor for that year})/12.$$

The factors include:

1. The optional Civilian Factors file (section E.2, below)
2. A factor that can be specified on the Control Inputs file (as described in Chapter II) to operate during certain (specified) calendar years, for all industries.

If more than one of these factor sets might apply for a given year, set 2 takes priority, over set 1. If neither of these factor sets apply, unity (1.0) is the *de facto* factor value.

b. Record and Format Guide

```
ISYGNP, IEYGNP, INTGNP, ICIVYR, NINDF          ( *)
┌for IND=1, NIND
└  IND, (GNPY(IYR, IND), IYR=ISYRC, IEYRC)      ( *)
```

c. Definitions of Symbolic Entities

*GNPY(IYR,IND) = Civilian demands (in \$M) on industry IND in year IYR.

*ICIVYR = Index year for the civilian demand data. All such data must be for the same year, and year must be the same as IDOLYR, as specified in the Element database file.

*IEYGNP = Ending year of data for civilian demand. Must equal IEYEAR, as specified in the Element database file.

IEYRC = Index for ending year of civilian demand data, relative to overall starting year of databases. Defined by $IEYRC = IEYGNP - ISYEAR + 1$, where ISYEAR is as read in from the Element database. (Since IEYGNP is required to equal IEYEAR, IEYRC is equal to the number of years of data in all databases.)

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

*INTGNP = Number of intervals in a year for civilian demand data (assumed to be 12 months). Not currently used.

*ISYGNP = Starting year of data for civilian demand. Must equal ISYEAR, as specified in the Element database file.

ISYRC = Index for starting year of civilian demand data, relative to overall starting year of databases. Defined by $ISYRC = ISYGNP - ISYEAR + 1$, where ISYEAR is as read in from the Element database. (Since ISYGNP is required to equal ISYEAR, ISYRC is equal to 1.)

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element database file.

4. Supply-Side Data on Q/K Ratios and Capacity Utilization Rates

a. Summary of Data in File

This file contains data on:

- The Q/K ratios, which are used FORCEMOB's modeling of investment
- Capacity utilization rates, which affect the amount of supply expansion possible for a given level of capital

In previous versions of FORCEMOB, these data appeared in the same file as the production, import, and export data. In Version 3, they have been placed in a separate file. These data probably will not change over the course of an analysis, but a user might well wish to examine different sets of production, import, and export data. The use of two separate input files facilitates this process.

Note: As used in FORCEMOB, the QKRATIO value corresponds to the output that could be produced in a *month*, per unit of capital in place. One can think of this value as being one-twelfth of an output/capital ratio as commonly defined.

The way that FORCEMOB treats the capacity utilization rates is explained in Volume I, Chapter II, section B.2, of this paper.

b. Record and Format Guide

ISYSUP, IEYSUP, NINDF (*)

COMMENT "CAPITAL/OUTPUT RATIOS" (A)

for IND=1, NIND
IND, QKRATIO(IND) (*)

COMMENT "CAPACITY UTILIZATION FRACTIONS" (A)

for IND=1, NIND
IND, (FFORCY(IYR, IND), IYR=ISYRS, IEYRS) (*)

c. Definitions of Symbolic Entities

*FFORCY(IYR, IND) = Ratio of peacetime production level to EOC level, for industry IND in year IYR (accounts for shift factor and capacity utilization rate).

IEYRS = Index for ending year of supply side data, relative to overall starting year of databases. Is equal to the number of years (IEYEAR-ISEYEAR+1) specified in the Element database; data must be provided for all these years.

*IEYSUP = Ending year of data for supply capabilities. Must equal IEYEAR, as specified in the Element database file.

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

ISYRS = Index for starting year of supply side data, relative to overall starting year of databases. Is equal to 1, as data must be provided for all years (ISEYEAR through IEYEAR) specified in the Element database.

*ISYSUP = Starting year of data for supply capabilities. Must equal ISEYEAR, as specified in the Element database file.

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element database file.

*QKRATIO(IND) = Monthly output in industry IND per unit of capital in place. Can be regarded as one twelfth of the traditional output/capital ratio for industry IND

(this latter ratio being estimated by the total output of industry IND divided by its capital stock).

5. Supply Database File

a. Summary of Data in File

The Supply database contains the major portion of the base (i.e., peacetime) supply-side data. It includes base supply (domestic production) capabilities by year and industry, along with employment (not currently used), imports, and exports. Base supply capability refers to the level of domestic production consistent with peacetime levels of capacity utilization. It should be consistent with the following assumptions:

- An associated capacity utilization level consistent with the value in the Capacity Utilization file
- A "standard peacetime" (but not explicitly specified) work week length

The methodology for supply-side modeling used by Version 1 of FORCEMOB differs considerably from the current methodology, and thus Version 1 has a somewhat different set of supply-side inputs, as discussed in [4]. Also, in Versions 1 and 2, data were input for each month and industry. In Version 3, data are input on a yearly basis; the code divides yearly values by 12 to obtain monthly values.

b. Record and Format Guide

ISYSUP, IEYSUP, ISUPYR, NINDF (*)

COMMENT "INDUSTRY BASE CAPABILITIES" (A)

for IND=1, NIND
IND, (SUPPLY(IYR, IND), IYR=ISYRS, IEYRS) (*)

COMMENT "PEOPLE EMPLOYED" (A)

for IND=1, NIND
IND, (PEOPLY(IYR, IND), IYR=ISYRS, IEYRS) (*)

COMMENT "IMPORTS" (A)

for IND=1, NIND
IND, (IMPRTY(IYR, IND), IYR=ISYRS, IEYRS) (*)

COMMENT "EXPORTS" (A)

for IND=1, NIND
IND, (EXPRTY(IYR, IND), IYR=ISYRS, IEYRS) (*)

c. Definitions of Symbolic Entities

*EXPRTY(IYR,IND) = Total exports (in \$M) in industry IND during year IYR.

IEYRS = Index for ending year of supply side data, relative to overall starting year of databases. Is equal to the number of years (IEYEAR-ISYEAR+1) specified in the Element database; data must be provided for all these years.

*IEYSUP = Ending year of data for supply capabilities. Must equal IEYEAR, as specified in the Element database file.

*IMPRTY(IYR,IND) = Imports (in \$M) in industry IND in year IYR. Real-valued variable.

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

*ISUPYR = Index year for the supply capabilities data. All such data must be for the same year, and this year must be the same as IDOLYR, as specified in the Element database file.

ISYRS = Index for starting year of supply side data, relative to overall starting year of databases. Is equal to 1, as data must be provided for all years (ISYEAR through IEYEAR) specified in the Element database.

*ISYSUP = Starting year of data for supply capabilities. Must equal ISYEAR, as specified in the Element database file.

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element database file.

*PEOPLY(IYR,IND) = Employment level in industry IND in year IYR. (Values must be input, but are not currently used. They will be used in a future version of FORCEMOB that considers labor constraints.)

*SUPLY(IYR,IND) = Amount (in \$M) of domestic supply produced in industry IND in year IYR, before any investment has been made.

6. Investment Distribution File

The way Version 3 of FORCEMOB models investment is considerably different from that of Version 1. Thus not surprisingly, the input investment data files of Version 3 differ a great deal from the Version 1 file described in [4]. Instead of one investment-related data file, there are three, and the types of data, while somewhat similar in spirit, differ considerably in detail. (The investment methodology of Version 2 is identical to that of Version 3, but there are some differences in file naming.)

a. Summary of Data In File

The Investment Distribution file is the largest of the three investment-related files. It gives the "investment demands" on industry—those demands on industries $\{j\}$ that are generated during the process of increasing capacity in industry i . (To increase capacity in a given industry will in general require contributions from many different industries.) As with most of its inputs, FORCEMOB expects these demands to be expressed in total requirements terms (see Volume I, Chapter II). That is, the investment demand should encompass not only the products and services purchased by the investment amount, but also all goods necessary to produce those products and services.

It is possible for several different industries to induce the same pattern of investment demand. To reflect this fact, and to reduce the size of the file, the investment

distribution data are organized into "investment distribution patterns." Suppose that a given industry, IND, has the distribution pattern JPAT, as specified on the Investment Sector Mapping data file (section 8, below). For any industry that has this investment distribution pattern, a dollar of investment in that industry induces an investment demand $CMAT(JIND,JPAT)$ on industry JIND, defined for each industry JIND from 1 through NIND.

Note that the value $CMAT(JIND, JPAT)$ represents a total amount over time. The FORCEMOB computer program divides this by the investment lead time of the industry IND (in which capacity is being built); the quotient represents the investment demand on industry JIND in each month of the investment lead time. The same divisor—the lead time of industry IND—is used for each "feeder" industry JIND. (The investment lead times are specified in the Investment Lead Times file, described in section 7, below.)

Not all industries need be listed for each distribution pattern; those not listed will be assigned an investment demand of zero. The user should check the file and make sure that such a zero value is appropriate. The number of different patterns should not exceed the value of the symbolic constant LNPAT, which is currently set to 57 in the code.

(In FORCEMOB Version 2, the Investment Distribution file had a structure identical to the current one. However, its name was hard-coded as CMAT.DAT, and the file was assumed to reside in the directory from which the program was being run.)

b. Record and Format Guide

```

┌for I=1,...(until end of file)
└ JIND, JPAT, CMAT(JIND,JPAT)                                     (*)

```

c. Definitions of Symbolic Entities

* $CMAT(JIND,JPAT)$ = If a given industry (say industry IND1) follows investment distribution pattern JPAT, then a dollar of investment in industry IND1 induces $CMAT(JIND,JPAT)$ investment demand in industry JIND. JIND ranges over the whole set of industries. The investment pattern JPAT is specified in the Investment Sector Mapping file.

JIND = Index of industry for which information is being specified on the current data line.

JPAT = Index for investment distribution pattern. Possible values range from 1 through the symbolic constant value LNPAT (currently set in the code at 57). See the

definitions of CMAT (above) and ICAPIND (in the Investment Sector Mapping file).

7. Investment Lead Times File

a. Summary of Data In File

The Investment Lead Times file gives, for each industry, the amount of time, in months, necessary to build additional productive capacity in that industry. Do not confuse the investment lead times with the Major End Item production lead times used in the Requirements module. The investment lead time does not depend on how much additional capacity is being built. Investment demand on all feeder industries is assumed to be spread out evenly over the investment lead time, as discussed in section 6.

As explained in Chapter II, the Control Inputs file contains a percentage factor that can be used to adjust the lead time values in the file. The same factor is used for all industries. FORCEMOB multiplies the file value by the factor, rounds to the nearest integer, and uses the resultant value as the investment lead time (with a minimum lead time of one month). One common interpretation is that the file has peacetime (greenfield) lead times, and the percentage factor in the Control Inputs file can reduce these lead times, in concordance with a mobilization scenario. A factor greater than 100 percent can be used, however, to examine the effect of lengthening the lead times.

Not all industries need be listed on the file; those that are not will be assigned an investment lead time of one month. The user should check the file and make sure that such a value is appropriate.

(In FORCEMOB Version 2, the Investment Lead Times file had a structure identical to the current one. However, its name was hard-coded as GREEN.DAT [GREEN for greenfield], and the file was assumed to reside in the directory from which the program was being run.)

b. Record and Format Guide

```
for I=1,...(until end of file)
  IND, GREEN1(IND) (*)
```

c. Definitions of Symbolic Entities

*GREEN1(IND) = Input lead time for investment in industry IND.

IND = Index of industry sector being considered on current line of the data file.

8. Investment Sector Mapping File

a. Summary of Data In File

The Investment Sector Mapping file contains, for each industry, the (index number of the) investment distribution pattern associated with investment in that industry. This pattern is used in determining which data in the Investment Distribution file to use for computing investment demand (as explained in section 6, above).

Technically, not all industries need be listed in this file. But beware—FORCEMOB assumes that any industry not listed induces zero investment demand. A value of zero for the pattern has the same effect. Also, to avoid errors, no value should exceed the symbolic constant LNPAT, the maximum number of patterns, which is currently set in the computer code at 57.

(In FORCEMOB Version 2, the Investment Sector Mapping file had a structure identical to the current one. However, its name was hard-coded as CAPIND.MAP, and the file was assumed to reside in the directory from which the program was being run.)

b. Record and Format Guide

```
┌for I=1,...(until end of file)
| IND, ICAPIND(IND)
└
```

(*)

c. Definitions of Symbolic Entities

*ICAPIND(IND) = Index of investment distribution pattern for industry IND, i.e., industry IND follows investment distribution pattern ICAPIND(IND).

IND = Index of industry for which information is being specified on the current data line.

E. OPTIONAL FORCEMOB INPUT FILES

1. Optional File 1—Base Military Factors

a. Summary of Data in File

The first optional file contains factors for base military requirements; the factors are multiplied by the values in the Base Military Requirements file (section D.1) to obtain the base military requirements values that FORCEMOB will use. A different factor value is specified for each industry and year combination, for each year within a specified range of years, not necessarily the whole scenario period. A factor for a given year applies to all months in that year. As indicated in section D.1, FORCEMOB allows several other sets of factors to modify the base military requirements. The user should be aware of the precedence order of those factors.

In Version 1 of FORCEMOB, the Base Military Requirements file could be organized either by Major End Item or by industry (as discussed in section D.1), and, similarly, the Base Military Factors file could be organized either way. Now, only the industry organization is used, for both files.

b. Record and Format Guide

NITEM, ISY, IEY, IFACTYPE

(*)

```
for IND = 1, NIND  
  IND, (FBMIND(IYR,IND), IYR=IYS, IYE) (I4, 19X, 15 (F7.4))
```

c. Definitions of Symbolic Entities

*FBMIND(IYR,IND) = Factor by which to multiply input base military requirements to obtain base military requirements actually used, for industry IND in year IYR. (Variable FBMIND is used if IBASTYPE=1, as it must be in the current code version. Factors CFACT(1) and PCFACT(1) are also applied to the base military requirements.)

IBASTYPE = Way in which base military demand is specified: =0, by MEI; =1, by industry. (In current code version, IBASTYPE must equal 1.)

*IEY = Ending year of data on file.

*IFACTYPE = Value must equal that of IBASTYPE, as specified in the Base Military Requirements database. For convenience, the definition of IBASTYPE appears above. (In current code version, IFACTYPE and IBASTYPE must equal 1.)

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

*ISY = Starting year of data on file.

IYE = Defined by $IYE = IEY - ISYEAR + 1$, where ISYEAR is as specified in the Element database. Last year of data, relative to overall starting year.

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

IYS = Defined by $IYS = ISY - ISYEAR + 1$, where ISYEAR is as specified in the Element database. First year of data, relative to overall starting year.

NIND = Number of industries (sectors)

NITEM = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element file.

2. Optional File 2—Civilian Factors

a. Summary of Data in File

The second optional file contains factors for civilian requirements; the factors are multiplied by values in the Civilian Requirements database file to get civilian requirements values that FORCEMOB will use (see section D.3). The factor values are given by year for each industry, for a specified range of years, not necessarily the whole scenario period. One use of these factors is to model civilian austerity during and/or after the conflict period.

b. Record and Format Guide

```
NINDF, ISY, IEY                                (*)
└for IND = 1, NIND
  IND, (FACCIV(IYR,IND), IYR=IYS,IYE)           (I4,19X,15(F7.4))
```

c. Definitions of Symbolic Entities

*FACCIV(IYR,IND) = Factor by which user wishes to multiply civilian demand on industry IND in year IYR.

*IEY = Ending year of data on file.

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND.
Dimension limit is the symbolic constant LNIND.

*ISY = Starting year of data on file.

IYE = Defined by $IYE = IEY - ISYEAR + 1$, where ISYEAR is as specified in the Element database. Last year of data, relative to overall starting year.

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

IYS = Defined by $IYS = ISY - ISYEAR + 1$, where ISYEAR is as specified in the Element database. First year of data, relative to overall starting year.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element file.

3. Optional File 3—Import/Export Factors

a. Summary of Data in File

The third optional file contains factors for imports and exports; the factors are multiplied by base values to get actual import and export values. The file works in conjunction with the import and export data found in the Supply database to adjust their levels (see section D.5). Factor values are by industry and year, for a specified range of years, not necessarily the whole scenario period.

b. Record and Format Guide

```
NINDF, ISY, IEY                                (*)

[for IND = 1, NIND
  IND, (FACIMP(IYR,IND), IYR=IYS,IYE)           (I4,19X,15(F7.4))
]

[for IND = 1, NIND
  IND, (FACEXP(IYR,IND), IYR=IYS,IYE)           (I4,19X,15(F7.4))
]
```

c. Definitions of Symbolic Entities

*FACEXP(IYR,IND) = Factor by which user wishes to multiply exports for industry IND in year IYR.

*FACIMP(IYR,IND) = Factor by which user wishes to multiply imports for industry IND in year IYR.

*IEY = Ending year of data on file.

IND = Index of industry sector. Ranges from 1 through the input limit variable NIND. Dimension limit is the symbolic constant LNIND.

*ISY = Starting year of data on file.

IYE = Defined by $IYE = IEY - ISYEAR + 1$, where ISYEAR is as specified in the Element database. Last year of data, relative to overall starting year.

IYR = Index for relative year. Used as an index for several variables that contain factors for year of the simulation. Ranges depend on specific use. Dimension limit is the symbolic constant LNYEAR.

IYS = Defined by $IYS = ISY - ISYEAR + 1$, where ISYEAR is as specified in the Element database. First year of data, relative to overall starting year.

NIND = Number of industries (sectors).

NINDF = Number of industries on the file under consideration. Value must equal NIND, as specified in the Element file.

4. Optional File 4—Major End Item Requirements

The MEI Requirements file may be used to specify demand for Major End Items directly—as opposed to having this demand computed by the military simulations of the FORCEMOB Requirements module (via the Force Structure database file). As indicated in Chapter II, calling the MEI Requirements file an “optional” file is somewhat misleading. Indeed, it is not required. But if the user requests it, then the whole structure of the Control Inputs file—and the FORCEMOB modeling—is affected. Nonetheless, this file is invoked in the same manner as the other optional data files.

The MEI Requirements file should not be confused with the MEI *Inventory* file (section C.2). The MEI inventories can be used to satisfy some or all of the requirements.

a. Summary of Data in File

This file contains total conflict requirements either by Major End Item and month or by Major End Item only. A flag at the top of the file indicates the organization of the

data in the file: IREQFLG=1 if requirements are given by MEI and month; IREQFLG=2 if they are given by MEI only. (IREQFLG=0 means that this file is not used.)

If IREQFLG = 2, not every Major End Item need be listed in the file. There is assumed to be no requirement for MEIs not listed.

As indicated in Chapter II (run options 0b and 1b), the value of IREQFLG must also appear on the Control Inputs file (if an MEI Requirements file is being used). If the value of IREQFLG is 2, then there must be an additional line on the Control Inputs file, to give the distribution of the MEI demand over the months of the conflict period.

b. Record and Format Guide

The first line always contains values for the variables IREQFLG, NMEIF, and IREQYR. The subsequent format depends on the value of IREQFLG. Complete formats for both cases are given below.

Format if IREQFLG = 1

```
IREQFLG,NMEIF,IREQYR          (*)
NMON2(1)                      (*)
COMMENT  "MEI REQUIREMENTS BY MONTH"  (A)
┌for IYR=1,NYR4
│  IYR                          (*)
│  ┌for IM = 1, NMEI
│  │IM, (DMDMEI(IMC,IM), IMC=IPST4,IPEND4)  (*)
│  └
└
```

Format if IREQFLG = 2

```
IREQFLG,NMEIF,IREQYR          (*)
COMMENT  "TOTAL MEI REQUIREMENTS"  (A)
┌for I = 1,...(until end of file)
│  IMF, TOTDMD(IMF)            (*)
└
```

c. Definitions of Symbolic Entities

*DMDMEI(IMC,IM) = Matrix of dollar value total conflict requirements for kind-IM Major End Item in relative month of battle IMC. Units = \$K.

IM = Index for kind of Major End Item. Ranges from 1 through the input limit variable NMEI. Dimension limit is the symbolic constant LNMEI.

IMC = Index for month of MEI demand, or equivalently, month of conflict period in theater 1. Theater 1 is used as a surrogate theater in which to accumulate the MEI demands. IMC ranges from 1 through NMON2(1), inclusive.

IMF = Index of MEI for which information is being specified on the current data line.

IPEND4 = Defined as $\min\{IPST4+11, NMON2(1)\}$, for a given value of IPST4. Index for ending month of the IYR-th year of MEI demand data.

IPST4 = Defined as $(IYR-1)*12 + 1$, for a value of IYR between 1 and NYR4, inclusive. Index for starting month of the IYR-th year of MEI demand data.

*IREQFLG = Flag to indicate organization of data. Value of 1 for demand by MEI and month (use DMDMEI); value of 2 for total demand by MEI only (use TOTDMD).

*IREQYR = Dollar year in which MEI requirements amounts are expressed. Must equal IDOLYR, as specified in the Element database file.

IYR = Relative year of MEI demand data within the conflict period in theater 1. Theater 1 is used as a surrogate theater in which to accumulate the MEI demands.

NMEI = Number of kinds of Major End Items.

NMEIF = Number of kinds of Major End Items on the file under consideration. Value must equal NMEI, as specified in the Element database file.

*NMON2(ITHR) = Initial specification of number of months of conflict in theater ITHR. Value on Control Inputs file for number of months of conflict cannot exceed NMON2. Here, only the value for ITHR=1 is input and used.

NYR4 = Defined as the smallest integer greater than or equal to $NMON2(1)/12$. Number of years that the MEI demand data encompass.

*TOTDMD(IM) = Total dollar demand during conflict period for Major End Items of kind IM. Units in \$K.

5. Optional File 5—Inventory Allocation

a. Summary of Data in File

This file contains the fractional distribution of MEI inventory over theaters, for each kind of Major End Item. It can be used to allocate initial MEI inventories to theaters for use in the military simulations of the FORCEMOB Requirements module. The inventories themselves are specified in the MEI Inventory database file; the optional

Inventory Allocation file contains only allocation factors. The factors are not percentages, but actual proportions (fractions).

The user should be aware of the following points.

1. The file gives allocation values for all *possible* theaters. The number of possible theaters is given by the symbolic constant LNTHR, which is set in the code to 4.
2. In a given run of FORCEMOB, not all of the theaters need be played. No renormalization of the file values is performed to account for theaters not played. The user must make sure that the Inventory Allocation file is compatible with the theaters played (e.g., that the file specifies zero allocation to theaters not played, or that some amount of inventory is to be withheld from the played theaters).
3. For a given MEI, the sum over theaters of the proportions need not be 1.0. The effect will be to withhold inventory or increase the amount of inventory from the value in the MEI Inventory file.
4. Not all MEIs need be listed on the optional Inventory Allocation file. The inventory allocation pattern specified on Control Inputs file is used for those MEIs that do not appear on the Inventory Allocation file.⁵
5. Conversely, for all MEIs listed in the Inventory Allocation file, the pattern in the Inventory Allocation file supersedes the pattern specified on the Control Inputs file.

b. Record and Format Guide

```
NMEIF                                     (*)
┌for I = 1,...(until end of file)
│  IMF, (EIINV(IMF, ITHR), ITHR=1, LNTHR)  (*)
└
```

c. Definitions of Symbolic Entities

*EIINV(IM, ITHR) = Fraction of inventory of MEI IM allocated to theater ITHR. EIINV(IM, ITHR) is set to THRINV(ITHR) for all IM unless values for EIINV are explicitly input by the user. The fraction THRINV corresponds to the percentage value specified on the Control Inputs file.

IMF = Index of MEI for which information is being specified on the current data line.

⁵ If the optional Inventory Allocation file is not requested, the allocation pattern specified on the Control Inputs file is used for all MEIs. See Chapter II and section C.2 of the current chapter for more information about this allocation.

ITHR = Index of theater. Ranges from 1 through the symbolic constant LNTHR.

LNTHR = Maximum number of theaters played (symbolic constant). Currently set to 4.

NMEI = Number of kinds of Major End Items.

NMEIF = Number of kinds of Major End Items. Value must equal NMEI, as specified in the Element file. (Used as a consistency check.)

6. Optional File 6—Military/Civilian Fungibility Factors

a. Summary of Data in File

Military/civilian fungibility, or dual use, factors are used to model the interchangeability of military vs. civilian productive capacity. These factors affect the supply expansion computation, as discussed in Volume I, Chapter II, section B.2 of this paper. If this optional file is not selected, then all industries are assumed to have complete fungibility.

Two different formats of the file are possible, as specified by the indicator MTHMCF on the second line of the file. In the first format, one value is used for all industries. In the second, a separate value can be specified for each industry. All factor values must be between zero (no fungibility) and 1.0 (complete fungibility), inclusive. In the second format, industries not listed are given a factor value of 1.0.

Versions 1 and 2 of FORCEMOB did not model dual use; they assumed complete fungibility for every industry. Version 3.0 could use fungibility factors, but the data were not formally treated as an optional file—instead, they were read in from a special auxiliary file with a hard-coded name.

b. Record and Format Guide

The first line of the file is a comment line, and the second line always contains a value for the variable MTHMCF. The subsequent format depends on the value of MTHMCF. Complete formats for both cases are given below.

Format if MTHMCF = 1—use one value for all industries

COMMENT "MIL/CIV FUNGIBILITY FACTORS FILE"	(A)
MTHMCF	(*)
COMMENT "SINGLE FUNGIBILITY FACTOR"	(A)
FGFMC	(*)

Format if MTHMCF = 2 (or any value not equal to 1)—let values vary by industry

COMMENT "MIL/CIV FUNGIBILITY FACTORS FILE"	(A)
MTHMCF	(*)
COMMENT "FUNGIBILITY FACTORS BY INDUSTRY"	(A)
for I = 1,...(until end of file)	
JIND, FRACMIL(JIND)	(*)

c. Definitions of Symbolic Entities

*FRACMIL(IND) = Military/civilian fungibility factor to be applied to industry IND. Value can range between zero (no fungibility) and 1.0 (complete fungibility). Industries not specified in the file are assigned a value of 1.0.

*FGFMC = Military/civilian fungibility factor to be applied to all industries. Value can range between zero (no fungibility) and 1.0 (complete fungibility). If MTHMCF = 1, FGFMC is read and then all elements of FRACMIL are set to FGFMC.

JIND = Index of industry for which information is being specified on the current data line.

*MTHMCF = Method for treating military/civilian fungibility.

MTHMCF = 1 — Use one factor value (FGFMC) for all industries.

MTHMCF = 2 (or any value not equal to 1) — Read different factor values (FRACMIL(IND)) for the various industries. Industries not on the list are assigned a factor value of 1.0 (complete fungibility).

Note: If Optional File 6 is not used, the result is equivalent to MTHMCF = 1 and FGFMC=1.0.

F. FORCEMOB AUXILIARY FILES

1. The Debugging Flags File

a. Summary of Data in File

This short file is intended to aid in program debugging, should this be necessary. It contains the values of certain special inputs to FORCEMOB that operate as "Debugging Flags." If these flags are set to certain values, then the model will print out

the values of certain computed variables. The definitions in section c show the specific values and uses of the flag variables.

This input file is optional. During execution, the program looks for a file named DEBUG.FLG in the directory from which the program is being run. If such a file exists, the program opens and reads the file. Otherwise, execution merely continues, and the flag variables are treated as though they have zero values.

Most of the values in the file were operative in Version 1 of FORCEMOB; now, most of them are not.

b. Record and Format Guide

The Debugging Flags file should have six records. FORCEMOB ignores the first, third, and fifth records, which can contain informative comments. The second record contains values for the variables ISUPTAB, IFLAG1, and IFLAG2, in 6I10 format (the extra fields in the format specification are not used). The fourth record contains values for the first five elements of vector IDBG (i.e., (IDBG(J),J=1,5)), again in 6I10 format. The sixth record contains (IDBG(J),J=6,10)), also in 6I10 format. The following schematic diagram encapsulates this information.

COMMENT "VARIABLES ISUPTAB, IFLAG1, IFLAG2"	(ignored)
ISUPTAB, IFLAG1, IFLAG2	(6I10)
COMMENT "VARIABLES IDBG(I), I=1, 5"	(ignored)
(IDBG(I), I=1, 5)	(6I10)
COMMENT "VARIABLES IDBG(I), I=6, 10"	(ignored)
(IDBG(I), I=6, 10)	(6I10)

c. Definitions of Symbolic Entities

*IDBG(J) = Flag to get special debug output from subroutine J. This output contains values of selected variables computed in subroutine J, and/or other informative messages. The output for subroutine J is generated only if IDBG(J) = 1; the output goes to the history file (except for J=6). The subroutines J are as follows. J=1,...,5—not used. J=6—investment routines; output goes to special file with extension .DGP. J=7—Subroutine INVNTY, J=8—Requirements module calculation routines, J=9—Subroutine REQSIM. The output for J=8 and J=9 is sketchy.

*IFLAG1 = Not currently used.

*IFLAG2 = Not currently used.

*ISUPTAB = Not currently used.

2. The Major End Item Aggregation Mapping File

a. Summary of Data in File

This auxiliary file is used if output reports 20 or 24 are requested (see Chapter II). These reports involve aggregation of the MEI demand by Service and budget categories, and this auxiliary file specifies the Service and budget categories associated with the various MEIs. The budget categories are shown in Table III-1. The Service categories are indexed as 1=Army/Marine Ground Equipment, 2=Navy/Marine Air and Sea, 3=Air Force. The Service and budget categories are (currently) hard-coded within the FORCEMOB computer program; they are not specified on a data file.

As stated earlier, the file must be named AGGMAP.DAT and must reside in the data file directory specified on the Control Inputs file. If FORCEMOB cannot find the file, it prints a message, does not generate the output report, and continues operation.

When the list of Major End Items (specified in the Element database file) changes, this auxiliary file might need to be changed also, to maintain compatibility of the mappings.

Table III-1. Budget Categories for MEI Aggregation

Budget Cat. No.	Budget Category Name
1	Milpers
2	O & M
3	Milcon/family housing
4	RDT&E
5	Aircraft
6	Missiles
7	Tanks
8	Other WTCV
9	Ammunition
10	Miscellaneous support equipment
11	Ships

Not all MEIs need be listed in the file; those not listed are simply not included in the aggregate values shown in the output report. The reading of the file ends when an end of file is encountered.

b. Record and Format Guide

```
for ITEM = 1, ...  
  MEIMAP ( ITEM ) , MAPAGG ( 1 , ITEM ) , MAPAGG ( 2 , ITEM ) , FACT ( ITEM )      ( I4 , 2 I5 , F10.6 )
```

c. Definitions of Symbolic Entities

*FACT(ITEM) = Fraction (between 0 and 1, inclusive) giving the proportion of demand for the specified MEI that is associated with the specified budget and Service category combination.

ITEM = Line in the AGGMAP.DAT file currently being considered.

*MAPAGG(K,ITEM) = Entry for K=1 gives the index of the budget category with which the Major End Item being considered in the current line of the AGGMAP.DAT file is associated. Entry for K=2 gives the index of the Service category for that MEI.

*MEIMAP(ITEM) = The (index of the) Major End Item being considered in the current line of the AGGMAP.DAT file.

REFERENCES FOR VOLUME II

- [1] White, Richard H., *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume I, *Theoretical Foundations*. IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.
- [2] Flythe, Mary Catherine, Timothy T. Fitzpatrick, Tara Santmire, Eleanor L. Schwartz, and Richard H. White, *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume II, *Users Guide*. IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.
- [3] White, Richard H., Eleanor L. Schwartz, Mary Catherine Flythe, and Timothy T. Fitzpatrick, *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume III, *Programmers Guide*. IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.
- [4] Schwartz, Eleanor L., and Richard H. White, *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume IV, *Data Preparation Guide*. IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.
- [5] Thomason, James S., D. Sean Barnett, Tara E. Santmire, Eleanor L. Schwartz, An-Jen Tai, and Richard H. White, *National Defense Stockpile Program Phase II: Assessments and Methodological Enhancements*. IDA Paper P-2885, Institute for Defense Analyses, Alexandria, VA, October 1993.
- [6] Thomason, James S., Eleanor Schwartz, An-Jen Tai, Sean Barnett, and David Graham, *IDA Analyses for the Naval Logistics 2001 Wargame '94*. IDA Paper P-2954, Institute for Defense Analyses, Alexandria, VA, May 1994.
- [7] McCarthy, Margaret Buckler, "LIFT: INFORUM's Model of the U.S. Economy." *Economic Systems Research*, Vol. 3, No. 1, 1991, pp.15-36.
- [8] Almon, Clopper, "The INFORUM Approach to Interindustry Modeling." *Economic Systems Research*, Vol. 3, No. 1, 1991, pp. 1-7.
- [9] *DOM: INFORUM's Detailed Output Model*. Interindustry Economic Research Fund, Inc., College Park, MD, October 1991.
- [10] *Microsoft FORTRAN PowerStation Professional Development System*, Version 1.0. Microsoft Corporation, Redmond, WA, 1992.

- [11] Santmire, Tara E., *Stockpile Sizing Module Documentation for Version 4.0*. Volume I, *Theoretical Foundations and Users Guide*, Volume II, *SSM Inputs*, and Volume III, *SSM Outputs*. IDA Paper P-2867 (draft), Institute for Defense Analyses, Alexandria, VA, June 1994.
- [12] Frazier, Thomas P., Carol G. Campbell, and Richard T. Cheslow, *The Revised Defense Translator*. IDA Paper P-2141, Institute for Defense Analyses, Alexandria, VA, October 1989.

GLOSSARY

ASCII	American Standard Code for Information Interchange
DEC	Digital Equipment Corporation
DEIMS	Defense Economic Impact Modeling System
DoD	Department of Defense
DOM	Detailed Output Model
DOS	Disk Operating System (Microsoft)
EOC	Emergency Operating Capacity
FEMA	Federal Emergency Management Agency
FM	Forces Mobilization Model (FORCEMOB)
FORCEMOB	Forces Mobilization Model
GDP	Gross Domestic Product
IDA	Institute for Defense Analyses
ILM	Industry-level Module (of FORCEMOB)
INFORUM	Inter-industry Forecasting at the University of Maryland
\$K	Thousands of Dollars
LIFT	Long-Term Inter-industry Forecasting Tool
\$M	Millions of Dollars
MEI	Major End Item
NDS	National Defense Stockpile
O&M	Operation and Maintenance
OSD	Office of the Secretary of Defense
PC	Personal Computer

Q/K Ratio	Capital/Output Ratio
RDT&E	Research, Development, Test, and Evaluation
REQMOD	Requirements Module (of FORCEMOB)
SIC	Standard Industrial Code
SSM	Stockpile Sizing Module (part of JIMPP)
TOE	Table of Organization and Equipment
VMS	Virtual Memory System
WTCV	Weapons and Tracked Combat Vehicles

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1996		3. REPORT TYPE AND DATES COVERED
4. TITLE AND SUBTITLE Documentation of the Forces Mobilization Model (FORCEMOB) Versions 3.1 and 3.2, Volume II: Data Preparation Guide			5. FUNDING NUMBERS C-DASW01-94-C-0054 TA- T-AO6-656	
6. AUTHOR(S) Eleanor L. Schwartz, James S. Thomason				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 1801 N. Beauregard Street Alexandria, VA 22311			8. PERFORMING ORGANIZATION REPORT NUMBER IDA Paper P-2953	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of the Assistant Secretary of Defense (Economic Security) The Pentagon, Room 2A318 Washington, DC 20301			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper provides documentation of Versions 3.1 and 3.2 of the Institute for Defense Analyses' Forces Mobilization Model (FORCEMOB). FORCEMOB is a quantitative model of the effect of an extraordinary military demand, such as a conflict or reconstitution, on the industrial base of the United States. It computes requirements for weapons and consumables that arise from a user-specified mobilization and conflict scenario that can last several years and can involve all Services and up to four theaters. Alternatively, time-phased military requirements can be input. These requirements are converted to demands on industry, which are added to any other military and civilian industrial demand. The model then compares total demand with industry supply and notes shortfalls, if any. The process of capacity expansion and investment to redress shortfalls is modeled. FORCEMOB accounts for the impact of time; most input data can vary by year, and lead times for production and investment are considered. The model contains many adjustment parameters that can be varied to conduct sensitivity analyses easily. FORCEMOB has been used in assessments of the National Defense Stockpile and in other studies of the industrial base.				
14. SUBJECT TERMS Computer Documentation, Modeling, Wargame, Economic Forecasting, National Defense Stockpile, FORCEMOB, Input-Output Analysis, Simulation Modeling, Strategic and Critical Materials, Mobilization, Industrial Base Assessments, Investment			15. NUMBER OF PAGES 102	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	